ON SOME STRUCTURAL AND ALGORITHMIC ASPECTS OF CERTAIN SUBCLASSES OF PERFECT GRAPHS

DEPARTMENT OF MATHEMATICS
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

October, 1993

ON SOME STRUCTURAL AND ALGORITHMIC ASPECTS OF CERTAIN SUBCLASSES OF PERFECT GRAPHS

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of
DOCTOR OF PHILOSOPHY

by
BHAWANI SANKAR PANDA

to the

DEPARTMENT OF MATHEMATICS
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

OCTOBER, 1993

CERTIFICATE

It is certified that the work contained in the thesis entitled "ON SOME STRUCTURAL AND ALGORITHMIC ASPECTS OF CERTAIN SUBCLASSES OF PERFECT GRAPHS", by **Bhawani Sankar Panda**, has been carried out under my supervision and that this work has not been submitted elsewhere for the award of a degree.

Thesis Supervisor

Dr. S. P. Mohanty

Professor, Department of Mathematics

INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

U.P. INDIA 208 016

OCTOBER, 1993.

DEDICATION

TO

MY PARENTS

Smt. Umakanti Panda

And

Shri Dhaneswar Panda

ACKNOWLEDGEMENT

Prof. S. P. Mohanty has been more than a thesis supervisor to me. I express my deep sense of gratitude to him for his constant encouragement and excellent supervision. His valuable suggestions and active patience throughout the thesis work have helped in improving the presentation of the thesis. I am very much thankful to him for building in me a right kind of attitude towards research. His warmth, friendliness, and fatherly affection have undoubtedly made my association with him a cherished phase in my life.

I am extremely thankful to Prof. S.B.Rao, Stat-Math Division, I.S.I., calcutta for his numerous discussions, fruitful suggestions, and constant encouragement. I am thankful to Prof. C. E. Veni Madhaban and Dr. Ashok Subramanian, IISc Bangalore, for their discussions and suggestions.

I specially thank Prof. P.C.Das for his constant encouragement and affection. I express my gratitude to my teachers Prof. U.B. Tewari, Prof. R.K.S.Rathore, and Prof. S.K.Gupta for their stimulating and inspiring lectures. Also, I thank Dr. R.K. Ghosh , Dr. S.Saxena , Dr. P.Gupta, and Dr. A. Mukopadhyay, all from the Department of Computer Science and Engineering, for their useful lectures.

I owe a lot to my teacher Prof. S. Pattanaik of Sambalpur University for his inspiration in different critical moments in my life and for motivating many of his students including me for higher studies in Mathematics. I specially thank my teachers Prof. B.B.Panda and Dr. A.K.Mishra, alumni of Department of Mathematics, I.I.T. Kanpur, for inspiring me to join I.I.T.. I am thankful to Dr. C.K.Mohapatra not only for supervising my M.Phil. dissertation entitled "Enumeration of Trees" but also for his constant interest in my work and friendliness.

I thank Dr. Supriya P. Mohanty, Faculty, BITS Pilani, for her

friendliness and for her interest in my work.

It is a pleasure to thank my colleagues M. Jayateerth and S. Siva Gurunathan for their discussions. It is a great occasion to thank Bhisma, Debu, Nagarajan, Raghu, Chanduka, Gouranga, Dandpath babu, Pattanaik, Balram, Rangan, Sudarsan sir, Motida, Rina bhabi, Sundar Rajan, Pratibha, Ratna, and Padma who have in one way or the other made my stay at IIT pleasurable.

It is a great pleasure to thank my special friend and colleague S. Chandra Sekhar Rao for his company, brotherliness, and for sharing my good moments and bad moments thoroughly.

I am very much thankful to aunty, Mrs. Hemalata Mohanty, who never let me feel I was away from my mother and from my home.

Lastly I thank my parents, sisters, brother, and my sister-in-law for their love, support, and patience.

(Bhawani Sankar Panda)

October, 1993.

SYNOPSIS

Perfect Graphs are an active area of current research for their potential applications to many real-life problems and for their nice combinatorial structures as well. Berge's Strong Perfect Graph Conjecture (SPGC)(See[1,3]) is responsible to a great extent for the theoretical developments of perfect graphs. The algorithmic aspects of perfect graphs was intensified by a fundamental result due to Grotschel et al [4], who showed that the four classical graph optimization problems, i.e. the problem of finding (i) the chromatic number, (ii) the clique number (iii) the clique covering number, and (iv) the independence number, admit polynomial solutions in perfect graphs, even though they are NP-Complete for general graphs [2].

Though a lot of work has been done in perfect graphs and its various subclasses, fundamental problems such as the SPGC, the recognition status of perfect graphs, and fast algorithms to solve the four classical graph optimization problems in perfect graphs are yet to be solved.

Thus a deep understanding of perfect graphs in structural as well as algorithmic view point is needed.

In this thesis we study some structural and algorithmic aspects of certain subclasses of perfect graphs. In chapter 1, we present some definitions, introduce perfect graphs and its various subclasses, and give a brief introduction to the organization of the thesis.

It is well known that a family S admits forbidden subgraph characterization iff it is closed under vertex induced subgraphs. Finding out minimal forbidden subgraphs for a class closed under vertex induced

subgraphs, is a classical problem in graph theory. Unfortunately, very little is known about this problem. Many families of graphs including perfect graphs and its various subclasses are closed under vertex induced subgraphs. But the problem of finding out the minimal forbidden subgraphs is open for many classes of graphs. In fact the SPGC can be restated as " The odd holes and the odd antiholes are the only forbidden subgraphs for perfect graphs".

In chapter 2, we propose a unified approach to solve the problem of finding minimal forbidden subgraphs for certain subclasses of perfect graphs which are closed under vertex induced subgraphs. Using this approach we obtain the following results.

(i) All the forbidden subgraphs for DV graphs; (ii) All the forbidden subgraphs for UV-graphs except certain "bad minimal forbidden subgraphs"; (iii)Forbidden subgraph characterization for k-trees which are also UV graphs, and (iv) Forbidden subgraphs for those split graphs which are also path graphs.

We present new characterizations of interval graphs and proper interval graphs following the framework of Monma and Wei[7], and show that the forbidden subgraphs of these classes, which are already known, can be obtained using our approach. We characterize planar chordal graphs in terms of separated graphs (see [7]), and obtain the forbidden subgraphs for these graphs. We show that the separator Theorem for RDV graphs obtained by Monma and Wei[7] has a flaw. We modify this and make some contribution towards finding the forbidden subgraphs for RDV graphs.

In chapter 3, we study intersection graphs of edge disjoint paths in a tree i.e. CV-graphs and intersection graphs of vertex disjoint paths in a tree i.e. PV-graphs. We first present several characterizations of CV-graphs including the forbidden subgraph characterization. We also

present a linear time sequential and an NC parallel algorithm for recognizing CV-graphs and for constructing a CV-clique tree for a CV-graph.

We show that the characterization of PV-graphs due to Samy et al [9] is not correct. We then characterize PV-graphs following the framework of Monma and Wei[7]. The forbidden subgraphs of PV-graphs are obtained following the framework presented in chapter 2. We also present a polynomial algorithm for recognizing PV-graphs and for constructing a PV-clique tree for a PV-graph.

A family of graphs is said to be complete for a conjecture if the truth of the conjecture on this restricted family implies that the conjecture is true in general. A family of graphs is said to be valid for a conjecture if the conjecture is true for this restricted class of graphs.

Identifying valid classes and complete classes for SPGC is an active area of research. In chapter 4, we study SPGC. We show that the family S of regular graphs having a transposition in their automorphism groups is a complete family for SPGC. We next show that certain proper subclasses of S are also complete for SPGC. A proper subclass of S is shown to be valid for SPGC. We conclude this chapter by disproving two conjectures of Holton[5] on stable graphs; graphs whose automorphism groups necessarily contain a transposition.

In chapter 5, we study perfect elimination orderings (PEOs) of chordal graphs(see [3]). PEO plays an important role in designing efficient algorithms in chordal graphs including the chordal graph recognition. We propose three algorithms, namely local maximum cardinality search (LMCS), maximum cardinality breadth first search (MCBFS), and maximum cardinality depth first search(MCDFS). MCBFS and MSDFS are natural applications of BFS and DFS, respectively, and they run in linear time. LMCS is a generalization of MCBFS, MCDFS, and maximum cardinality search (MCS) of

Tarjan et al [10]. We show that none of the above four algorithms or the Lexicographic breadth first search (LEX-BFS) due to Rose et al [8] can generate any arbitrary PEO of an arbitrary chordal graph. We then compare these algorithms as far as generating any arbitrary PEO is concerned. Many graph problems including the four classical optimization problems that are hard for general graphs, can be solved in polynomial time in chordal graphs. However, testing Hamiltonicity, determining the domination number, etc. are NP-complete even for chordal graphs. We study the Hamiltonian problem in chordal graphs through Hamiltonian elimination ordering (HEO) (see [6]). We characterize H-Perfect Hamiltonian chordal graphs, i.e. Hamiltonian chordal graphs which have HEOs, and present a linear time algorithm for finding a Hamiltonian cycle of an H-Perfect chordal graph given an HEO in the input. We then propose an $O(n^2m)$ algorithm for recognizing and for constructing an HEO of H-Perfect chordal graphs. We characterize H-Perfect k-trees. We finally present linear time algorithms to construct HEOs of H-Perfect k-trees and of proper interval graphs, and thus linear time algorithms for finding Hamiltonian cycles in H-Perfect k-trees and proper interval graphs.

Some open problems and conjectures old and new relating to the topics discussed in the thesis are mentioned at appropriate places.

References

1. C.Berge and V. Chvatal (Eds.), Topics on Perfect Graphs, Ann. of Discrete Math. 21 (1984).

2. M.R.Garey and D.S.Johnson, Computers and Intractability: A guide to the theory of NP-completeness,(W.H.Freeman and Co, San francisco).

3. M.C.Golumbic, Algorithmic graph theory and perfect graphs, (Academic press,New york 1980).

4. M.Grotschel, L.Lovasz, and A.Schrijver, Polynomial algorithms for Perfect graphs, Ann. Discrete Math. 21 (1984) 325-356.

5. D.A.Holton, A report on stable graphs, J.Austral. Math. Soc. 15 (1973) 163-171.

6. R.E.Jamison and R.Laskar, Elimination orderings of chordal graphs, in: 'Proc. of the seminar on combinatorics and applications' (K.S.Vijain et al (ed.)) I.S.I. Calcutta, 1982, 192-200.

7. C.L.Monma and V.K.Wei, Intersection graphs of paths in a tree, JCT B 41 (1986) 141-181.

8. D.Rose, R.E.tarjan, and G.Luekar, Algorithmic aspects of vertex elimination on graphs, SIAM J. Comput. 5 (1976) 266-283.

9. A. N. Samy, G. Arumugam, M. Paul Devasahayam, and C. Xavier, A recognition algorithm for the intersection graphs of internally disjoint paths in a tree, in: P.S.Thiagarajan ed., National seminar on Theoretical computer science, (Madras, India, 1991) 169-178.

10. R.E.Tarjan and M.Yanakakis, Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs, SIAM. J. Comput.13 (1984) 225-331.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER - 1
INTRODUCTION

1.1 Introduction:

Many real-life problems can be modeled using Graph Theory. So there is a lot of interest in algorithmic Graph theory and thus in structural graph theory as behind many efficient algorithms there lie good structural characterizations. Unfortunately, many important computational graph theoretic problems are NP-Hard for general graphs[47]. However, graphs that arise in practice are not always arbitrary; they possess certain structures. For example, the class of perfect graphs that arises in many applications [11,26,39,57,58,110,134,136] admits polynomial solutions for several problems having practical significance. The concept of perfect graph, introduced by C.Berge in 1960 [5-7], has turned out to be one of the most rewarding in Graph Theory. The Strong Perfect Graph Conjecture (SPGC) (which will be stated in section 1.3), posed by C.Berge in early 1960s [6,7,11], is responsible to a great extent for the theoretical developments of Perfect graphs. Theorems on perfect graphs often generalize other important results in various fields of graph theory. Perfect graphs have also contributed to many problems including the theory of antiblocking polyhedra[44,45], the study of facial structure of polytopes (see [88]), and the theory of integer linear programming (see[11,88]). Grotschel et al [63,64] have shown that the four classical optimization problems, namely finding (1) clique number, (2) chromatic number, (3) independence number, and (4) clique covering number, can be solved in polynomial time for perfect graphs. This fundamental result has intensified the algorithmic interest in perfect graphs. In several special classes of perfect graphs

many optimization problems are solvable by fast algorithms which display important ideas of combinatorial algorithms (such as depth first search and greedy algorithm).

Perfect graphs and its subclasses are an active area of current research for their potential applications to different areas. Though a lot of work has been done in perfect graphs and its various subclasses, fundamental problems such as the SPGC, the recognition status of perfect graphs, and fast combinatorial algorithms to solve the four classical graph optimization problems remain to be solved.

Thus a deep understanding of perfect graphs in structural as well as algorithmic view point is needed.

In this thesis we study some structural and algorithmic aspects of perfect graphs and certain subclasses of perfect graphs.

In section 2, we introduce basic definitions and notations which will be used in this thesis. In section 3, we introduce perfect graphs, the SPGC, and some of the approaches to settle SPGC. In section 4, certain subclasses of perfect graphs, relevant to this thesis, are introduced.

We conclude this chapter with a brief overview of the organization of this thesis.

1.2 Definitions And Notations:

In this thesis we assign the number i.j to the jth section of the ith chapter and the number i.j.k to the kth result of the jth section of the ith chapter. Throughout this thesis, we use "iff" for if and only if, "wlg" for without loss of generality, "w.r.t." for with respect to, "s.t." for such that, "$\in$" for belongs to, and "$\notin$ " for does not belong to.

We always assume our graph G=(V,E) to be finite, undirected, simple, and connected unless otherwise mentioned. Our concepts and notations on graphs conform to those of Harary[67], Bondy and Murthy[16], and Berge[10].

Let $G[S]$, $S \subseteq V$, denote the induced subgraph on S. A set $C \subseteq V$ is said to be a clique if $G[C]$ is a complete subgraph of G. A clique of G that is not properly contained in any other clique of G is called a maximal clique of G. Let $\mathcal{C}(G)$ denote the set of maximal cliques of G, and $\mathcal{C}_v(G)$, $v \in V(G)$, denote the set of maximal cliques of G containing v. The clique number $\omega(G)$ of a graph is the size of a maximum clique in G. A set S is an independent set if $G[S]$ is a null graph. The independence number $\alpha(G)$ of G is the size of a maximum independent set of G. A proper coloring of G is to assign color to the vertices of G s.t. no two adjacent vertices receive the same color. The minimum number of colors needed for a proper coloring of G is called the chromatic number of G and is denoted by $\chi(G)$. A clique cover for G is a set of cliques of G whose union is V. The clique covering number $\theta(G)$ of G is the size of a minimum clique cover of G. The problem of finding (i) $\omega(G)$, (ii) $\alpha(G)$, (iii) $\theta(G)$, and (iv) $\chi(G)$ are known as the four classical graph optimization problems. We call C_{2k+1}, $k>1$, an odd hole and its complement $\overline{C_{2k+1}}$, $k>1$ an odd antihole.

Let $G_1=(V_1,E_1)$ and $G_2=(V_2,E_2)$ be any two graphs. A bijection β from V_1 to V_2 is said to be an isomorphism from G_1 to G_2 if $xy \in E_1$ iff $\beta(x)\beta(y) \in E_2$. An isomorphism for G onto G itself is called an automorphism. $\Gamma(G)=\{\beta$ s.t. β is an automorphism of $G\}$ is the automorphism group of G. A permutation on a set that interchanges two elements and fixes the remaining elements is called a transposition. We denote a transposition β that interchanges x and y by $\beta=(x\ y)$. S_n denotes the symmetric group on n symbols. Let $S_p(q)$ denote the permutation group isomorphic to S_p but q fold in nature. For group theoretic concepts we refer to Weilend[140] and Harary[67].

Let G_S, $S \subseteq V$, denote $G-S$. Let $\Gamma(G)_S$ denote the subgroup of $\Gamma(G)$ which fixes S. An ordering $\alpha=(v_1,v_2,\dots,v_n)$ of V is said to be a stabilizing

sequence of G if $\Gamma(G_{S_j}) = \Gamma(G)_{S_j}$, $1 \le j \le n$, where $S_j = \{v_1, v_2, \ldots, v_j\}$. A stable graph is a graph with a stabilizing sequence. The concept of stability was introduced and extensively studied by Holton[69-72].

For definitions and concepts regarding sequential algorithm used in this thesis, we refer to Golumbic[57], Aho et al[1], Horowitz et al[73], Knuth[82], and Garey and Johnson[47]. For basic graph algorithms we refer to Even[38], Gibbons[54], Christofides[23], Mchugh[90], Golumbic[57], Deo [33], and Aho et al[2]. For the notion of NP-completeness we refer to Garey and Johnson[47], and Golumbic[57].

In this thesis we propose some parallel algorithms as well. Our formal model of parallel computation is the Parallel random access machine (PRAM), proposed by Fortune and wyllie[42]. In this shared memory(SM) model, in a unit of time each processors can read a unit of data, write it, or perform an elementary arithmetic or logical operation on it. The number of bits in each data element is bounded by a logarithmic function of the number of processors.

Several different types of PRAMS arise which differ in how simultaneous read and write access to the same location by different processors are resolved. The exclusive-read exclusive-write(EREW) PRAM forbids simultaneous reading from or writing to the same location. The concurrent-read exclusive-write(CREW) PRAM model allows two processors to read from the same location, but not to write it. In the strongest model, the concurrent-read concurrent-write(CRCW) PRAM model, two processors may simultaneously read from or write to the same location. There are several different ways of resolving the case when two processors try to write different values to the same location. In the 'arbitrary' CRCW model write conflicts to a location are resolved by choosing one of the values written to the location, the choice of which is arbitrary. In this thesis by CRCW

PRAM model we mean the 'arbitrary' CRCW PRAM model unless stated otherwise.

A problem is said to be in the complexity class NC if there is a PRAM algorithm that uses a number of processors polynomial in the size of the problem and its running time is bounded by a polynomial in the logarithm of the problem size. Theoretical computer scientists consider NC to be the class of problems with efficient parallel algorithms. Cook[29] gives a survey of the complexity class NC. In practice, what is efficient depends on the actual parallel architecture used because PRAMs are only an abstract model. Also the class NC only measures those algorithms that are efficient in asymptotic sense.

For other definitions, concepts, and for basic parallel graph algorithms we refer to Quinn[105], Akl[3], Gibbons and Rytter[55], Quinn et al[104], and Moitra et al[93].

1.3 Perfect Graphs And SPGC:

For any graph G,

$$\omega(G) \le \chi(G) \quad \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

and

$$\alpha(G) \le \theta(G) \quad \ldots\ldots\ldots\ldots\ldots\ldots (2).$$

These inequalities are dual to one another since $\alpha(G)=\omega(\overline{G})$ and $\theta(G)=\chi(\overline{G})$. A graph G has the property (P1) if $\chi(G[S])=\omega(G[S])$ for every $S \subseteq V$, and has the property (P2) if $\alpha(G[S])=\theta(G[S])$ for every $S \subseteq V$. A graph G satisfying property (P1) (property (P2)) is said to be χ-perfect(α-perfect). A graph G is perfect if it is α-perfect and χ-perfect. It is clear by duality that a graph G is α-perfect iff $\overline{G}$ is χ-perfect.

Graphs satisfying $\alpha(G)=\theta(G)$ played an important role in Shannon's 1956 paper[117]. In that paper he remarked that the smallest graph G with $\alpha(G) < \theta(G)$ is C_5.

It was shannon's work which motivated Berge to make a conjecture concerning graphs that satisfy inequality (2) with equality sign. This conjecture is known as Strong Perfect Graph Conjecture (SPGC) and can be stated in many equivalent forms, one of which is the following:

" A Graph G is α-perfect iff it does not contain odd holes and odd antiholes as induced subgraphs."

Thinking that this conjecture might be too difficult Berge made the following weak perfect graph conjecture. "A graph G is α-perfect iff it is χ-perfect." Weak perfect graph conjecture (WPGC) was cultivated by Fulkerson ([43-45]) and finally proved by Lovasz[86]. This result, now known as perfect graph theorem, has made the concept of α-perfectness and χ-perfectness obsolete. Therefore, here after by a perfect graph we mean χ-perfect, and by SPGC we mean the following: " A graph G is perfect iff it contains neither an odd hole nor an odd antiholes as an induced subgraph."

The SPGC is yet to be settled and challenges both, those who would like to see more and more advances in combinatorics and those who prefer the ingenious elementary proofs of 'classical' graph theory.

A family of graphs is said to be complete for a conjecture if the truth of the conjecture on this restricted family implies the truth of the conjecture in general. A family of graphs is said to be valid for a conjecture if the conjecture is true for this restricted class of graphs.

Identification of valid classes and complete classes for SPGC is an active area of research. Many special classes of graphs such as $K_{1,3}$ free graphs[98], torodial graphs[62], K_4-e free graphs[99], planar graphs[135] and many others[20,34,83,89,96,126,133,137] have been shown to be valid classes for SPGC.

D.G.Corneil [30] is probably the first researcher to identify many

classes of graphs to be complete for SPGC.

A class $\mathcal{F}_{\mathcal{S}}$ of graphs is said to be the minimal forbidden subgraphs for the class $\mathcal{S}$ iff (i) $H \in \mathcal{F}_{\mathcal{S}}$ implies $H \notin \mathcal{S}$ but $H[C] \in \mathcal{S}$ for every $C \subset V(H)$, and (ii) $G \in \mathcal{S}$ iff G does not contain any induced subgraph isomorphic to a member of $\mathcal{F}_{\mathcal{S}}$. A family $\mathcal{S}$ admits forbidden subgraph characterization if there exists a class $\mathcal{F}_{\mathcal{S}}$ of minimal forbidden subgraphs for $\mathcal{S}$. It is well known that a family $\mathcal{S}$ admits forbidden subgraph characterization iff it is closed under vertex induced subgraphs. But finding out minimal forbidden subgraphs for a class closed under vertex induced subgraphs, is a very important problem. Unfortunately, there is no unified approach to solve this problem. Many families of graphs including perfect graphs and its various subclasses are closed under vertex induced subgraphs. But the problem of finding out the minimal forbidden subgraphs is open for many classes of graphs. In fact, the SPGC can be restated as " The odd holes and the odd antiholes are the only forbidden subgraphs for perfect graphs". So SPGC is a conjecture concerning the forbidden subgraphs for perfect graphs.

Many interesting classes of graphs are shown to be subclasses of perfect graphs (see[8,11,36,57,88]). For more about SPGC we refer to [9,11,25,57,87,88,91,138].

1.4 Subclasses Of Perfect Graphs:

A graph G=(V,E) is said to be chordal if every cycle in G of length at least four has a chord. Chordal graphs are also known as triangulated graphs, rigid circuit graphs, monotone transitive graphs, and perfect elimination graphs in the literature. Chordal graphs are extensively studied in recent years(see[18,21,35,36,40,51,57,58,108-113,118,122,132]. Chordal graphs were first analyzed by Hajnal and suranyi[65]. They showed that $\alpha(G)=\theta(G)$ for a chordal graph. Berge[5] proved that $\chi(G)=\omega(G)$ for a chordal graph G. In view of the perfect graph theorem any one of these

results imply that chordal graphs are perfect, since any induced subgraph of a chordal graph is chordal. Chordal graphs have many applications in areas such as evolutionary trees[18], archaeology[19] facility location[22], scheduling[97], and solutions of sparse systems of linear equations[112].

A subset $S \subset V$ is a u-v vertex separator for nonadjacent vertices u and v if u and v lie in different connected components of G-S. If no proper subset of S is a u-v separator, then S is a **minimal** u-v separator. A minimal vertex separator S of G is a minimal u-v separator for some u and v. Chordal graphs can be characterized in terms of minimal u-v separator as follows:

Theorem 1.4.1: [35] G is chordal iff every minimal u-v separator is a clique.

A vertex v of G is a simplicial vertex if G[N(v)] is a clique in G, where N(v) denotes the neighbors of v. An ordering $\alpha=(v_1,v_2,\dots,v_n)$ of V is a perfect elimination ordering (PEO) of G if v_i is a simplicial vertex of $G_i=G[\{v_i,v_{i+1},\dots,v_n\}]$, $1\le i\le n$.

Dirac[35] and then Lekkerkerker[84] proved the following result.

Theorem 1.4.2: ([35,84]) A chordal graph G has a simplicial vertex. Moreover, if G is non-complete, then G has two nonadjacent simplicial vertices.

Using induction and Theorem 1.4.2, the following characterization of chordal graphs can be obtained.

Theorem 1.4.3: [46] G is chordal iff G has a PEO. Moreover, any simplicial vertex can be the starting vertex of some PEO of G.

The concept of PEO is very important in chordal graphs. It turned out that all the existing chordal graph recognition algorithms and many graph optimization problems, including the four classical ones, in chordal graphs

make use of the PEO. So PEOs of chordal graphs have been extensively studied by researchers and various algorithms have been developed for their construction (see[112,119,128,129]). There are two linear time algorithms, namely Maximum cardinality search (MCS) due to Tarjan et al [129], and Lexicographic breadth first search (LEX-BFS), due to Rose et al [114] to construct PEOs. LEX-BFS makes use of certain lexicographic orderings and turns out to be a breadth first search (BFS).

Other types of elimination orderings have also been investigated and subclasses of chordal graphs have been characterized in terms of these elimination orderings.

Let $\alpha=(v_1,v_2,\dots,v_n)$ be any ordering of V. We denote $(v_n,v_{n-1},\dots,v_1)$ by α^{-1}. Let $N^+[v_i]=\{v_j$ s.t. $v_j=v_i$ or $i<j$ and $v_iv_j \in E(G)$ $\}$ and $N^-[v_i]=\{v_j$ s.t. $v_j=v_i$ or $i>j$ and $v_iv_j \in E(G)\}$, $1\le i\le n$.

α is called an interval elimination ordering (IEO) iff for each i, $N^-[v_i]$ is an interval in the ordering α.

A PEO α is said to be a Strong elimination ordering (SEO) if $i<j<k<l$, and v_iv_k, v_iv_l, and $v_jv_k \in E(G)$, then $v_jv_l \in E(G)$. A graph having an SEO is called a strongly chordal graph. A PEO α is called a Bicompatible elimination ordering (BCO) iff α^{-1} is also a PEO of G. A PEO $\alpha=(v_1,v_2,\dots,v_n)$ is called an Hamiltonian elimination ordering (HEO) if $v_1,v_2,\dots,v_n$ is a Hamiltonian cycle of G. A graph having an HEO is called an H-Perfect chordal graph.

A graph T_n is called a trampoline if it consists of $K_n=\{v_1,v_2,\dots,v_n\}$ together with a set of n independent vertices $\{u_1,u_2,\dots,u_n\}$ s.t. for each i, u_i is adjacent to only v_i and v_j, where $j=i-1(\text{mod } n)$.

Strongly chordal graphs have been studied extensively by M.Farber[40] and several characterizations have been obtained by him. Strongly chordal graphs admit the following forbidden subgraph characterization.

Theorem 1.4.4: [40] G is strongly chordal (equivalently, G has an SEO) iff G does not contain any C_n, $n \ge 4$ and any trampoline T_n as induced subgraphs.

Before presenting characterizations of graphs in terms of BCO and IEO, we introduce some more concepts.

Let F be a finite family of nonempty sets. An undirected graph G is an intersection graph for F if there is a one to one correspondence between the vertices of G and the sets in F s.t. two vertices in G are adjacent if the two corresponding sets have nonempty intersection. If F is a family of intervals in a linearly ordered set (like the real line), then G is called an interval graph. If no interval of F properly contains the other set-theoretically, then G is called a proper interval graph. If each interval in F is of unit length, then G is called a unit interval graph. It is well known[109] that proper interval graphs are exactly the unit interval graphs.

Interval graphs arise in many application areas and have been extensively studied by researchers. Interval graphs can be recognized in linear time (see[17,79]). For problems and applications of interval graphs we refer to [57,58,110].

Every maximal clique Q_i of an interval graph G corresponds to a point q_i in the real line s.t. $q_i \in \cap\{ I_x$ s.t. $x \in Q_i\}$, where I_x is the interval that corresponds to the vertex x. Let $Q_1, Q_2, \ldots, Q_r$ be an ordering of the maximal cliques of an interval graph G s.t. the points $q_1, q_2, \ldots, q_r$ corresponding to the cliques is in increasing order. Then the ordering $Q_1, Q_2, \ldots, Q_r$ satisfies the following property. If $v_i \in Q_j \cap Q_k$, then $v_i \in Q_s$ for every s, $j \le s \le k$. This property of interval graphs turns out to be a characterizing property.

Theorem 1.4.5[56] An undirected graph G is an interval graph iff the maximal cliques of G can be linearly ordered s.t. for each vertex x of G,

the maximal cliques containing x occur consecutively.

Interval graphs admit the following forbidden subgraph characterization.

Theorem 1.4.6: [84] A graph G is an interval graph iff it does not contain any of the graphs in Figure 1.4.1 as an induced subgraph.

Proper interval graphs have the following forbidden subgraph characterization.

Theorem 1.4.7: [109] An interval graph G is a proper interval graph iff it does not contain $K_{1,3}$ as an induced subgraph.

Now we state the characterizations of graphs having IEO and BCO, respectively.

Theorem 1.4.8: [76] A graph G has an IEO iff it is an interval graph.

Theorem 1.4.9: [76] A graph G has a BCO iff it is a proper interval graph.

The k-trees, introduced by Harary and Palmer[68], are an important subclass of chordal graphs. They have been characterized by Beineke and Pippert[4], and subsequently by Rose[113] in many interesting ways. Furthermore, the k-trees have been extensively investigated by Proskurowski[101-103], and Foata[41]. We give below the definitions and various characterizations of k-trees.

A graph G is a k-tree if it can be obtained by the following recursive rules.

(a) Start with any k-clique as the basis graph. A k-clique is a k-tree.

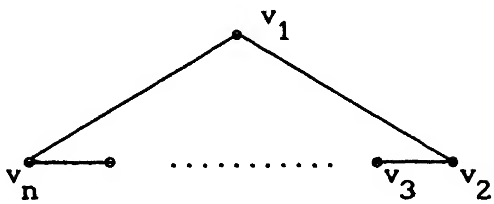
(b) To any k-tree H add a new vertex and make it adjacent to a k-clique of H, to form a (k+1)-clique.

Theorem 1.4.10: The following are equivalent.

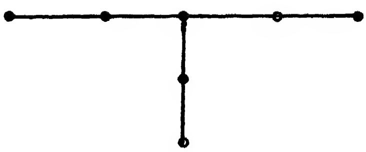
(1) G=(V,E) is a k-tree.

(2) [113] (a) G is connected,

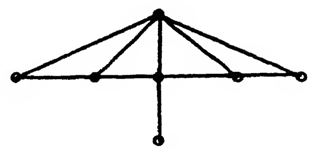
(b) G has a k-clique but no (k+2)-clique, and



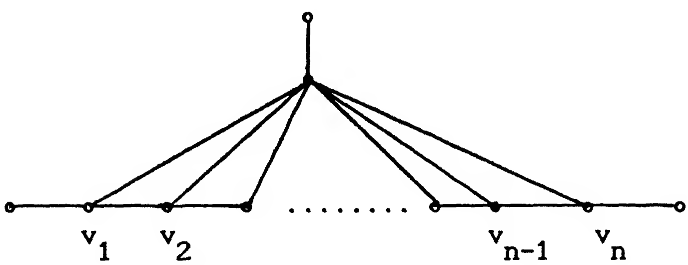
$I_1 (n \geq 4)$



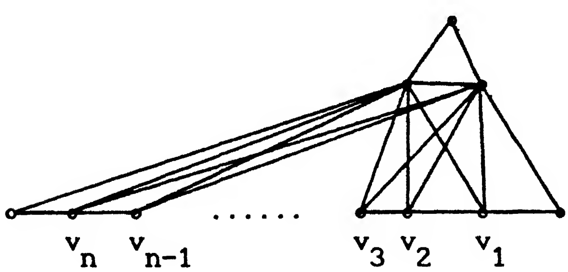
I_2



I_3



$I_4 (n \geq 2)$



$I_5 (n \geq 3)$

Figure 1.4.1: Forbidden Subgraphs for Interval Graphs.

(c) every minimal vertex separator of G is a k-clique.

(3) [113] (a) G is chordal,

(b) $|E|=k|V|-k(k+1)/2$, and

(c) G has a k-clique but no (k+2)-clique.

(4) [113] (a) G is connected,

(b) $|E|=k|V|-k(k+1)/2$, and

(c) every minimal vertex separator of G is a k-clique.

Let $\mathcal{B}(G)$ denote the set of blocks, i.e. the biconnected components of G. The block graph of G is the intersection graph of $\mathcal{B}(G)$, where the intersection of two blocks means the intersection of their vertex sets. A graph G is said to be a block graph if it is isomorphic to the block graph of some graph. Harary[66] introduced the notion of block graphs and proved, among other results, that a graph G is a block graph iff each of its blocks is complete. Clearly block graphs are a subclass of chordal graphs.

We next discuss about intersection graphs of subtrees in a tree with specified properties.

Let T be any tree and F be any collection of subtrees of T. Two subtrees T_1 and T_2 in F are said to have nonempty intersection if $V(T_1) \cap V(T_2) \neq \phi$. The intersection graph of subtrees in a tree is called a subtree graph. Walter[139], Gavril[51], and Buneman [18] have shown that the subtree graphs are exactly the chordal graphs. In fact, they proved the following result.

Theorem 1.4.11: [18,51,139] The following statements are equivalent.

(1) G is chordal.

(2) G is the intersection graph of a family of subtrees in a tree.

(3) There exists a tree T s.t. $V(T)=\mathcal{C}(G)$, and $F=\{T[\mathcal{C}_v(G)],\ v \in V(G)\}$ is a collection of subtrees in T.

The pair (T,F) satisfying Theorem 1.4.11(3) is called a clique tree

representation for the chordal graph G. This representation has motivated the researchers to study other types of intersection graphs. Renz[108] introduced path graphs as the intersection graphs of paths in a tree. Path graphs have been extensively studied by Gavril[53] and Renz[108]. The intersection graph of directed paths in a rooted directed tree, known as directed path graph has been introduced and extensively studied by Gavril[52]. Later, Monma and Wei[92] introduced a unified framework for the study of a variety of intersection graphs that arise in the context of paths in trees. Six different classes of intersection graphs can be defined in this context as follows. A path is said to be a vertex (edge) path if the path is considered to be the set of vertices(edges) making up the path. A graph G is an undirected(directed) vertex path graph or UV(DV) graph if it is the intersection graph of a family of undirected (directed) vertex paths in an undirected (directed) tree. A DV graph with a rooted tree representation is called a rooted directed vertex graph or RDV graph. UV graphs are nothing but path graphs introduced by Renz[108], and RDV graphs are nothing but directed path graphs. In stead of vertex path if we take edge path, we will get three different types of graphs, namely UE, DE, and RDE graphs. The UE graphs have been studied by Golumbic and Jamison [60], Lobb [85], Syslo [127], and Tarjan [130]. For applications of intersection graphs we refer to[58,59,78].

Two paths P_1 and P_2 are said to be vertex disjoint if either $V(P_1) \cap V(P_2) = \phi$, or $v \in V(P_1) \cap V(P_2)$ implies v is an end vertex of at least one of the paths P_1 and P_2. The intersection graph of a family of vertex (edge) disjoint paths in a tree T is said to be a perfect(compact) vertex graph or PV-graph(CV-graph). Samy et al[115] introduced the notion of PV-graphs and CV-graphs and characterized these graphs following the framework of Monma and Wei[92]. Unfortunately their characterization for PV graphs is not

correct (see chapter 3).

Though in the definitions of UV,DV, and RDV graphs the trees are arbitrary, there exist trees with nice properties, which are given in the following Theorem.

Theorem 1.4.12: (Clique Tree Theorem) [92] (a) A graph $G=(V,E)$ is a UV graph iff there exists a tree T with vertex set $\mathcal{C}(G)$, s.t. for every $v \in V(G)$, $T[\mathcal{C}_v(G)]$ is a path in T.

(b) A graph $G=(V,E)$ is a DV graph iff there exists a directed tree T with vertex set $\mathcal{C}(G)$, s.t. for every $v \in V(G)$, $T[\mathcal{C}_v(G)]$ is a directed path in T.

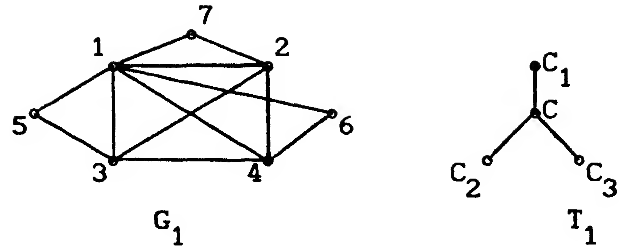
(c) A graph $G=(V,E)$ is an RDV graph iff there exists a rooted directed tree T with vertex set $\mathcal{C}(G)$, s.t. for every $v \in V(G)$, $T[\mathcal{C}_v(G)]$ is a directed path in T.

A tree satisfying Theorem 1.4.12 is called a clique tree for the graph it characterizes. In Figure 1.4.2, we give a chordal graph, a UV graph, a DV graph, and an RDV graph with their clique trees.

Next we present the characterizations of UV, DV, and RDV graphs due to Monma and Wei[92], in terms of separated subgraphs. To this end we need to introduce some new concepts.

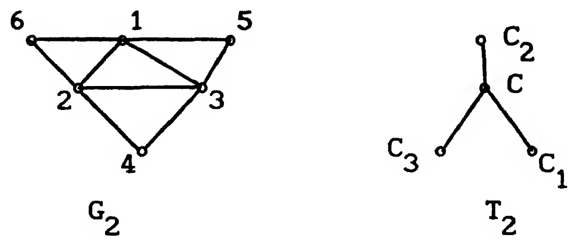
If $G-C$ is disconnected for a clique C with components $H_i=(V_i,E_i)$, $1 \le i \le r$, $r \ge 2$, then C is said to be a separating clique and $G_i=G[(V_i \cup C)]$, $1 \le i \le r$, $r \ge 2$, is said to be a separated graph of G w.r.t. C. A graph with no separating clique is called an 'Atom'. Let C be a separating clique of G. Cliques which intersect C but not equal to C are called relevant. In the following only relevant cliques are considered.

Let C_1 and C_2 be two cliques of G. We say that (1) C_1 and C_2 are unattached, $(C_1 | C_2)$ if $(C_1 \cap C) \cap (C_2 \cap C) = \phi$, otherwise they are attached, (2) C_1 dominates C_2, $(C_1 \ge C_2)$ if $C_1 \cap C \supseteq C_2 \cap C$, (3) C_1 properly dominates C_2, $(C_1 > C_2)$ if $C_1 \cap C \supset C_2 \cap C$, (4) C_1 and C_2 are congruent,



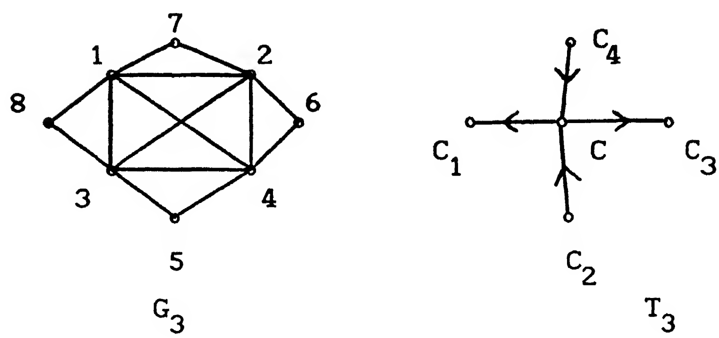
$C=\{1,2,3,4\}$
$C_1=\{1,2,7\}$
$C_2=\{1,4,6\}$
$C_3=\{1,3,5\}$

(i) A Chordal Graph G_1 and its clique tree T_1.



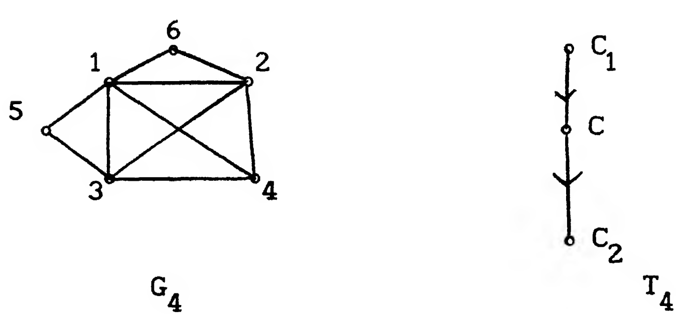
$C=\{1,2,3\}$
$C_1=\{\ 2,3,4\}$
$C_2=\{1,2,6\}$
$C_3=\{1,3,5\}$

(ii) A UV graph G_2 and its clique tree T_2.



$C=\{1,2,3,4\}$
$C_1=\{1,3,8\}$
$C_2=\{3,4,5\}$
$C_3=\{2,4,6\}$
$C_4=\{1,2,7\}$

(iii) A DV graph G_3 and its clique tree T_3.



$C=\{1,2,3,4\}$
$C_1=\{1,2,6\}$
$C_2=\{1,3,5\}$

(iii) An RDV graph G_4 and its clique tree T_4.

Figure 1.4.2: Examples of Intersection Graphs and their Clique Trees

$(C_1 \sim C_2)$ if they are attached and $(C_1 \cap C) = (C_2 \cap C)$, and (5) C_1 and C_2 are antipodal, $(C_1 \Leftrightarrow C_2)$ if they are attached and neither dominates the other.

Let G_1 and G_2 be two separated graphs of G w.r.t. C. We say that (1) G_1 and G_2 are unattached, $(G_1 | G_2)$, if $C_1 | C_2$ for every clique C_1 in G_1 and every clique C_2 in G_2, (2) G_1 dominates G_2, $(G_1 \geq G_2)$, if they are attached, and for every clique C_1 in G_1, either $C_1 \geq C_2$ for all cliques C_2 in G_2, or $C_1 | C_2$ for all cliques C_2 in G_2, (3) G_1 properly dominates G_2, $(G_1 > G_2)$, if $G_1 \geq G_2$ but not $G_2 \geq G_1$, (4) G_1 is congruent to G_2, $(G_1 \sim G_2)$, if G_1 dominates G_2 and G_2 dominates G_1; in this case $C_1 \sim C_2$ for every C_1 in G_1 and every C_2 in G_2, and (5) G_1 and G_2 are antipodal, $(G_1 \Leftrightarrow G_2)$ if they are attached and neither dominates the other. The relation 'congruent to' is an equivalence relation on the set S_G of separated graphs of G w.r.t. C(see[92]). The equivalence classes of S_G under this relation are called congruence classes. The above concepts were introduced by Monma and Wei [92].

The following Lemma due to Monma and Wei[92] will be used in chapters 2 and 3.

Lemma 1.4.13: Any collection of pairwise non-antipodal separated graphs of a (general) graph can be arranged in such a way that $G_i > G_j$ implies $i < j$.

For any separated subgraph G_i, let $W(G_i)$ be the set of $v \in C$ s.t. there is a vertex $w \in (V(G_i)-C)$ for which the edge $vw \in E(G_i)$. An antipodal pair $G_i \Leftrightarrow G_j$, w.r.t. C is said to be relevant to x if $x \in W(G_i) \cap W(G_j)$. Relevant cliques of G_i which contain $W(G_i)$ are said to be principal cliques of G_i.

Now we state the characterizations of UV,DV, and RDV graphs due to Monma and Wei [92], in terms of separated subgraphs.

Theorem 1.4.14: [92] (Separator Theorem) Assume that C separates $G=(V,E)$ into subgraphs $G_i=G[V_i \cup C]$, $1 \le i \le r$, $r \ge 2$.

(a) G is a UV graph iff each G_i is UV, and the G_i's can be colored s.t. no antipodal pairs have the same color, and for each $v \in C$, the set of subgraphs neighboring v is 2-colored.

(b) G is a DV graph iff each G_i is DV, and G_i's can be two colored s.t. no antipodal pairs have the same color.

(c) G is an RDV graph iff each G_i is RDV, and the G_i's can be 2-colored s.t. no antipodal pairs have the same color, and that in one color every subgraph has an RDV clique tree rooted at C, and that in the other color no two subgraphs are unattached and every subgraph (with one possible exception) has an RDV clique tree rooted at a relevant clique. The exceptional subgraph, should it exist, is dominated by every other subgraph of the same color, and it has an RDV clique tree in which the vertex C has out-degree zero.

1.5 Outline of the Thesis:

In chapter 2, we propose a unified approach to solve the problem of finding minimal forbidden subgraphs for certain subclasses of perfect graphs which are closed under vertex induced subgraphs. We then find the forbidden subgraphs for DV graphs. We introduce the notion of "bad forbidden subgraph" for UV graphs, and find out all minimal forbidden subgraphs for UV graphs except the "bad forbidden subgraphs". Our strong intuitive feeling motivates us to conjecture that there is no bad forbidden subgraphs for UV graphs. We then give some evidence to support our conjecture. It is shown that the forbidden subgraph characterizations for interval graphs and proper interval graphs can be obtained using our framework. Next we characterize chordal planar graphs using the frame work of Monma and wei[92], and obtain the forbidden subgraphs for this class. A

parallel recognition algorithm for chordal planar graphs is presented. We also study the forbidden subgraphs of RDV graphs and make some contributions towards the forbidden subgraph characterization for RDV graphs.

In chapter 3, we study PV-graphs and CV-graphs. We first present several characterizations of CV-graphs including the forbidden subgraph characterization. We show that CV graphs are exactly the block graphs. We also present a linear time sequential and an NC parallel algorithm for recognizing CV-graphs and for constructing a CV-clique tree for a CV-graph.

We present a counter example to the characterization of PV-graphs due to Samy et al [115]. Then we characterize PV-graphs following the framework of Monma and Wei[92]. The forbidden subgraphs of PV-graphs are obtained following the framework presented in chapter 2. We conclude this chapter with a polynomial algorithm for recognizing PV-graphs and for constructing a PV-clique tree for a PV-graph.

In chapter 4, we study SPGC. We show that the family S of regular graphs having no transposition in their automorphism groups is a complete family for SPGC. We next show certain proper subclasses of S are also complete for SPGC. A proper subclass of S is shown to be valid for SPGC. We conclude this chapter by disproving two conjectures of Holton[69] on stable graphs.

In chapter 5, we study PEOs of chordal graphs. As mentioned earlier, PEOs are useful in designing efficient algorithms in chordal graphs including the chordal graph recognition. We propose three algorithms, namely, maximum cardinality breadth first search (MCBFS), maximum cardinality depth first search (MCDFS), and local maximum cardinality search (LMCS). MCBFS and MCDFS are natural applications of BFS and DFS respectively, and they run in linear time. LMCS is a generalization of

MCBFS, MCDFS, and MCS. We then compare these algorithms as far as generating any arbitrary PEO is concerned. Many graph problems including the four classical optimization problems that are hard for general graphs, can be solved in polynomial time in chordal graphs. However, testing Hamiltonicity, determining the domination number, and other problems are NP-complete even for chordal graphs. We study the Hamiltonian problems in chordal graph through HEO. We characterize H-Perfect Hamiltonian chordal graphs,, and present a linear time algorithm for finding a Hamiltonian cycle of an H-Perfect chordal graph given an HEO in the input. We then propose an $O(n^2m)$ algorithm for recognizing and for constructing an HEO of H-Perfect chordal graphs. We then characterize H-Perfect k-trees. We finally present linear time algorithms to construct HEOs of H-Perfect k-trees and proper interval graphs, and thus linear time algorithms for finding Hamiltonian cycles in H-Perfect k-trees and proper interval graphs.

CHAPTER 2

ON FORBIDDEN SUBGRAPHS OF CERTAIN SUBCLASSES OF PERFECT GRAPHS

2.1 Introduction:

Since the class of perfect graphs is closed under vertex induced subgraphs, it admits forbidden subgraph characterization. In fact, Berge's conjecture says that odd holes and odd antiholes are the only forbidden subgraphs for perfect graphs, which is yet to be settled. Also for many other classes of graphs including UV,DV and RDV graphs, the problem of finding out forbidden subgraphs is open. Though for some classes of graphs, like bipartite graphs, split graphs, interval graphs, etc. forbidden subgraphs are known, the methods used to obtain them are ad hoc. Since many classes of graphs admit forbidden subgraph characterization, it is of significant interest to look for a general approach for tackling this problem.

In this chapter we suggest a unified approach for finding out the forbidden subgraphs for UV, DV, Interval, proper interval, and chordal planar graphs. The problem of finding the forbidden subgraphs for UV and DV graphs were open where as that for interval graphs and proper interval graphs are known(see [84,109]).

2.2 Preliminaries:

Let $\mathscr{C}_1$ be the class of DV graphs, $\mathscr{C}_2$ be the class of UV graphs, $\mathscr{C}_3$ be the class of interval graphs, $\mathscr{C}_4$ be the class of proper interval graphs, $\mathscr{C}_5$ be the class of chordal planar graphs, and $\mathscr{C}_6$ be the class of RDV graphs.

Note that $\mathcal{C}_i$, $1\le i\le 6$, is a subclass of chordal graphs. Let $\mathcal{G}$ = {$\mathcal{S}$ s.t. $\mathcal{S}$ is closed under vertex induced subgraphs}. A nontrivial forbidden subgraph for a class $\mathcal{S}$ is a forbidden subgraph with a separating clique.

Next we suggest a unified approach for finding out the nontrivial forbidden subgraphs for an arbitrary class $\mathcal{S}$ where $\mathcal{S} \in \mathcal{G}$.

Let $\mathcal{S}^*$ ={G $\in$ $\mathcal{S}$ s.t. G has a separating clique}. First characterize G $\in$ $\mathcal{S}$ in terms of separated graphs. This characterization is called the 'separator theorem' for the class $\mathcal{S}^*$. Let H be a nontrivial forbidden subgraph for $\mathcal{S}$. Then H has a separating clique. Now each separated graphs of H w.r.t. any separating clique belongs to $\mathcal{S}$. So the separated graphs of H w.r.t. any separating clique will violate some condition of the 'separator theorem' for the class $\mathcal{S}$. Then using this characterize H.

Unfortunately, this approach does not help us in obtaining the forbidden subgraphs for perfect graphs, since every forbidden subgraph for a perfect graph is trivial.

Since C_n, $n\ge4$ is the only trivial forbidden subgraph for $\mathcal{C}_i$, it is enough to find nontrivial forbidden subgraphs for $\mathcal{C}_i$, $1\le i\le 6$. To obtain them, we first need the 'separator Theorem' for $\mathcal{C}_i$, $1\le i\le 6$. Since the separator theorems for $\mathcal{C}_1$,$\mathcal{C}_2$, and $\mathcal{C}_6$ are there (Theorem 1.4.14), we will obtain the 'separator Theorem for $\mathcal{C}_3$,$\mathcal{C}_4$, and $\mathcal{C}_5$.

Let G be a chordal graph, and G_i, $1\le i\le r$, $r\ge2$ be the separated graphs of G w.r.t. some separating clique C of G. Then we have:

Proposition 2.2.1: Every separated graph G_i of a chordal graph G has a principal clique.

Proof: We induct on k, where k = $|V(G_i)-C|$. If k=1, then clearly $(W(G_i) \cup (V(G_i)-C))$ is a principal clique of G. Assume that k > 1. Clearly G_i is a non-complete chordal graph. So by Theorem 1.4.2, G_i has two non-adjacent simplicial vertices. Let w be a simplicial vertex of G_i s.t. w $\notin$ C. Then

G_1-w is again a separated subgraph of G w.r.t. C. So by induction hypothesis G_1-w has a principal clique, say C_1'. If $\{w\} \cup C_1'$ is a clique of G_1 then take $C_1= C_1' \cup \{w\}$, otherwise take $C_1= C_1'$. Then C_1 is a principal clique of G_1. Hence the proposition is proved by induction.∎

2.3 Forbidden Subgraphs for DV Graphs:

Let H_i, $1\le i\le r$, $r\ge 2$ be the separated graphs of a chordal graph H w.r.t. a separating clique C of H. Define the graph $\mathcal{A}(H,C)$ as follows. $V(\mathcal{A})= \{H_i$ s.t. H_i is a separated graph of H w.r.t. C$\}$ and $E(\mathcal{A})= \{H_iH_j$ s.t. $H_i \Leftrightarrow H_j\}$.

A separated graph of G G_i is said to be a strong separated graph if there exists an induced odd cycle $\alpha = G_1, G_2, ..G_i, ..., G_{2k+1}$, $k > 1$, of $\mathcal{A}(G,C)$ s.t. G_i dominates G_j for all j except $j=i-1$ and $j=i+1$. (operations on the indices are under modulo $(2k+1)$).

Lemma 2.3.1: Let $\alpha = G_1, G_2, ..., G_{2k+1}$, $k > 1$ be an induced odd cycle of $\mathcal{A}(G,C)$, where G is a chordal graph (not necessarily a critical DV graph). Then the following conditions hold.

(i) If G_i dominates some G_j, then G_i is a strong separated graph.

(ii) There exist at most two strong separated graphs. Moreover, if there are exactly two, then they appear consecutively in α.

Proof: (i) Wlg, i=1. Now $G_1 \ge G_j$. We claim that $G_1 \ge G_m$, $3\le m\le 2k$. Suppose G_1 does not dominate some G_r. Wlg, $3 \le r \le j-1$. Let r_1, $3\le r_1\le j-1$ be the largest index s.t. G_1 does not dominate G_{r_1}. Since $G_{r_1+1} \Leftrightarrow G_{r_1}$ and G_1 dominates G_{r_1+1}, G_1 is attached to G_{r_1}. If G_{r_1} dominates G_1, then G_{r_1} also dominates G_{r_1+1}, as G_1 dominates G_{r_1+1}, and the relation "domination" is a transitive relation on the set of separated graphs. Since $G_{r_1+1} \Leftrightarrow G_{r_1}$, $G_1 \Leftrightarrow G_{r_1}$, because by assumption, G_1 does not dominate G_{r_1}. Hence α is not a chordless cycle, which is a contradiction. So lemma 2.3.1(i) holds.

(ii) If possible let there be three strong separated graphs, say G_{i_1}, G_{i_2}, G_{i_3}. Since k>1, wlg we assume that G_{i_1} is not antipodal to G_{i_2}. So $G_{i_1} \sim G_{i_2}$ and G_{i_1} and G_{i_2} do not occur consecutively on α. Then the separated graphs G_{i_1}, $G_{i_1+1}, \ldots, G_{i_2-1}$ form a cycle. This contradicts the fact that α is a chordless cycle. So there are at most two strong separated graphs. Again if two such separated graphs exist, then they must occur consecutively on α; otherwise, using the similar analysis it can be shown that α is not a chordless odd cycle.∎

Let $G \in \mathcal{F}_{\mathcal{C}_1}$. Then G has at least three cliques. If G is not chordal, then it must be isomorphic to C_n, $n \geq 4$. Let G be chordal. So G has a separating clique as it has more than two cliques. Let C be a separating clique of G and let G_i, $1 \leq i \leq r$, $r \geq 2$ be the separated graphs.

The following lemma gives the structure of $\mathcal{A}(G,C)$.

Lemma 2.3.2: $\mathcal{A}(G,C)$ is isomorphic to C_{2k+1} for some $k \geq 1$.

proof: Since $G \in \mathcal{F}_{\mathcal{C}_1}$, each G_i is a DV graph. So by Theorem 1.4.14, G_i's cannot be 2-colored in such a way that antipodal pairs receive different colors. So $\mathcal{A}(G, C)$ is not bipartite and hence contains an odd cycle. Since a graph containing an odd cycle also contains an induced odd cycle, $\mathcal{A}(G,C)$ contains an induced odd cycle, say C_{2k+1}, for some k. Since $G \in \mathcal{F}_{\mathcal{C}_1}$, $\mathcal{A}(G,C)$ is isomorphic to C_{2k+1}.∎

We have seen that the concept of antipodality of separated graphs plays an important role in the structure of $\mathcal{A}(G,C)$ for $G \in \mathcal{F}_{\mathcal{C}_1}$. The following result due to Monma and Wei [92] characterize the antipodality of two separated graphs of an arbitrary graph.

Lemma 2.3.3 [92]: Two separated subgraphs G_1 and G_2 are antipodal iff

(1) $C_1 \Leftrightarrow C_2$, or

(2) $C_1 > C_2$, $C_1' < C_2'$, or

(3) $C_1 > C_2$, $C_1' \geq C_2'$, $C_1' \mid C_2''$ (or, $C_2 > C_1$, $C_2' \geq C_1'$, $C_2' \mid C_1''$), or

(4) $C_1 \sim C_2$, $C_1 \mid C_2'$, $C_2'' \sim C_1'$, $C_2'' \mid C_1''$

for some cliques C_1, C_1' , C_1'' in G_1 and C_2, C_2' , C_2'' in G_2. (These cliques need not all be different.)

However, in the following lemma we show that conditions (1) and (2) are necessary and sufficient for the antipodality of two separated subgraphs of a chordal graph.

Lemma 2.3.4: Two separated graphs G_1 and G_2 of a chordal graph G are antipodal iff (1) $C_1 \Leftrightarrow C_2$, or (2) $C_1 > C_2'$, $C_1' < C_2$, for some cliques C_1, C_1' in G_1 and C_2, C_2' in G_2.(Condition (2) is obtained from condition (2) of Lemma 2.3.3 by interchanging C_2 and C_2').

Proof: Since it is easy to see that either of the conditions (1) and (2) implies $G_1 \Leftrightarrow G_2$, we prove the necessity only.

Assume $G_1 \Leftrightarrow G_2$, and condition (1) is not satisfied. So no clique of G_1 is antipodal to any clique of G_2. Let C_i be a principal clique of G_i, $1 \leq i \leq 2$.

Case 1: $W(G_1) = W(G_2)$.

Then each of G_1 and G_2 has at least two relevant cliques. Again G_i, has a relevant nonprincipal clique, say, C_i', $1 \leq i \leq 2$; otherwise, one separated graph dominates the other. Now C_1, C_1', C_2, and C_2' satisfy the condition (2).

Case 2: $W(G_1) \neq W(G_2)$.

Wlg, let $W(G_1) \subset W(G_2)$. Since condition (1) is not satisfied, and G_2 does not dominate G_1, there exists a clique C_2' in G_2 s.t. C_2' is attached to C_1 and $W(G_1)-(C_2' \cap C) \neq \phi$. Since C_2' is not antipodal to C_1, $C_2' \cap C \subset W(G_1)$. So C_1, C_1, C_2, and C_2' satisfy the condition (2).∎

Since the antipodality of separated graphs plays an important role in the structure of critical DV graphs, we first characterize the antipodality in terms of forbidden subgraphs.

Lemma 2.3.5: Let G be a chordal graph having an antipodal pair w.r.t. some separating clique. Then G contains a subgraph isomorphic to one of the graphs in Figure 2.3.1.

Proof: Let C be a separating clique of G and (G_1, G_2) be an antipodal pair w.r.t. C.

Case 1: There exists C_i in G_i, i=1,2 s.t. $C_1 \Leftrightarrow C_2$.

Then clearly $|C| \geq 3$. Let $\{x,y,z\} \subseteq C$ be s.t. $x \in C_1 \cap C_2$, $y \in (C_1 \cap C)-C_2$, and $z \in (C_2 \cap C)-C_1$. Let $x_i \in C_i-C$, i=1,2. Then $G[\{x,y,z,x_1,x_2\}]$ is isomorphic to H'_1.

Case 2: No clique of G_1 is antipodal to any clique of G_2.

Subcase 2(a) $W(G_1) = W(G_2)$.

Then there exist C_1 and C'_1 in G_1, C_2 and C'_2 in G_2(all distinct) s.t., $C_1 > C'_2$ and $C_2 > C'_1$. Let T_i be a clique tree for G_i, i=1,2. Let $P=C,C^*_1(i),C^*_2(i),\ldots,C^*_r(i),C'_i$ be the path from C to C'_i in T_i. Wlg, $C^*_j(i)$ is a principal clique of G_i, for $1 \leq j \leq r$. Let $G'_i=G[\{C \cup C^*_r(i) \cup C'_i\}]$. Then clearly $G'_1 \Leftrightarrow G'_2$. So if $G_1 \Leftrightarrow G_2$, and $W(G_1)=W(G_2)$, then we may assume that each of G_1 and G_2 has exactly two relevant cliques.

Subcase 2(a.1): $(C'_2 \cap C \cap C'_1)=\phi$.

Let $x \in C'_2 \cap C$, $y \in C'_1 \cap C$, $x_1 \in (C_1 \cap C'_1)-C$, $y_1 \in (C_2 \cap C'_2)-C$, $x_2 \in C'_1-C_1$, $y_2 \in C'_2-C_1$, and $z \in C-W(G_1)$. Then $G[\{x,y,z,x_1,x_2,y_1,y_2\}]$ is isomorphic to H'_2.

Subcase 2(a.2): $(C'_2 \cap C \cap C'_1) \neq \phi$.

Let $x \in C'_2 \cap C \cap C'_1$, $y \in W(G_1)-C'_1$, $z \in C-W(G_1)$, $x_1 \in (C_1 \cap C'_1)-C$, $x_2 \in C'_1-C_1$, $y_1 \in (C_2 \cap C'_2)-C$, and $y_2 \in C'_2- C_2$. Then $G[\{x,y,z,x_1,x_2,y_1,y_2\}]$ will be isomorphic to H'_3.

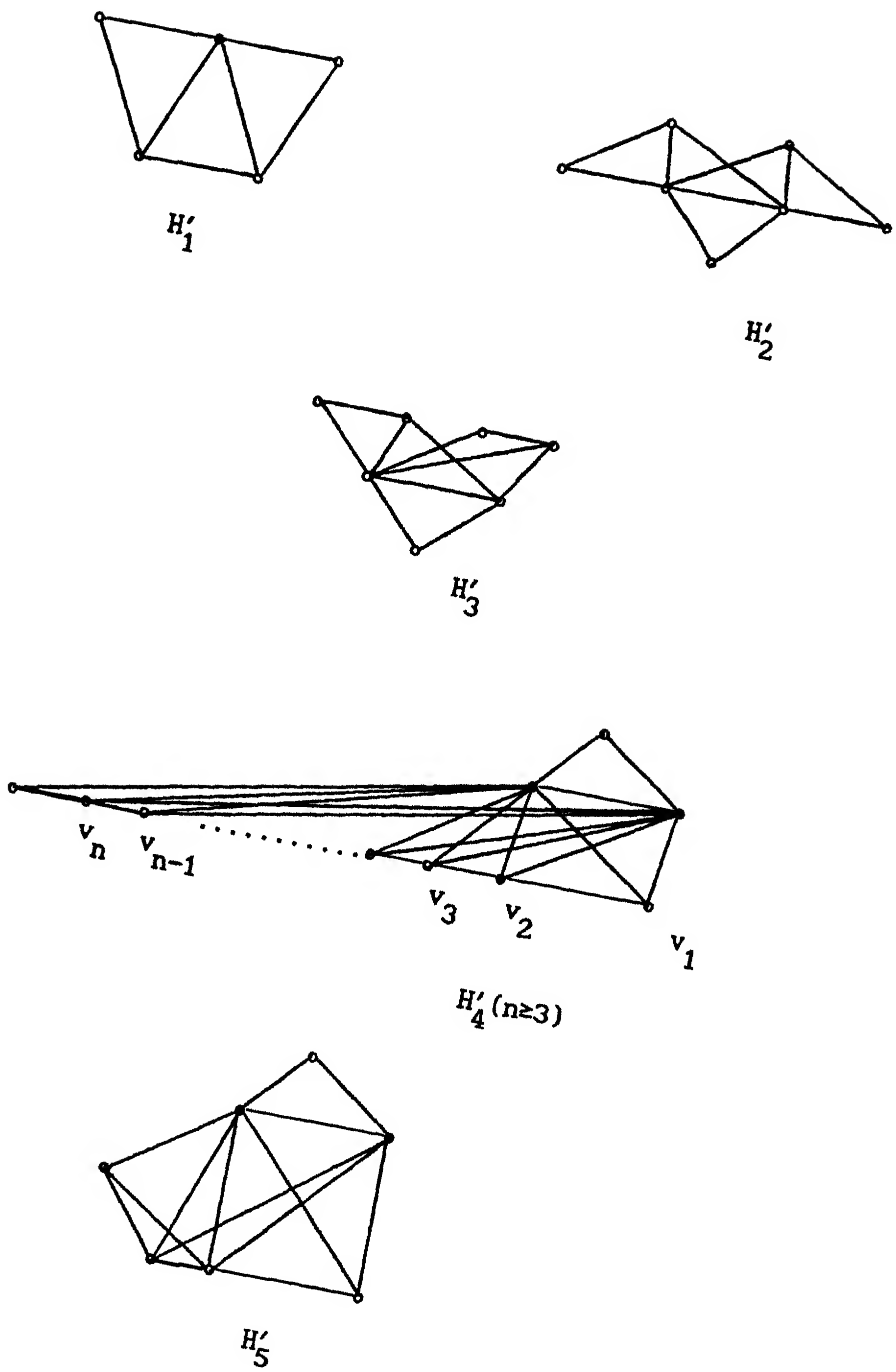


Figure 2.3.1: Antipodality In Terms of Forbidden Subgraph.

Subcase 2(b): $W(G_1)$ is a proper subset of $W(G_2)$.

Let C_i be a principal clique of G_i, $i=1,2$. Now by Lemma 2.3.4, there exists C_2' s.t. $C_1 > C_2'$. Since G_2 is chordal, G_2 has a clique tree T_2. Let $Q=C,C_2,C_3,\ldots,C_n,C_{n+1}$, where $C_{n+1}=C_2'$ be the path from C to C_2' in T_2. Clearly, $C_{i+1} \cap C \subset C_i \cap C$, $1 \le i \le n$. Wlg, assume that $C_i \ge C_1$, $2 \le i \le n$. Clearly, $|C| \ge 4$. let $G' = G[\{C,C_2,\ldots,C_{n+1}\}]$. Then G' is a chordal graph. So, by Theorem 1.4.2, G' contains at least two simplicial vertices. Since C and C_{n+1} are the only end vertices of the clique tree Q of G', there exist simplicial vertices z and v_1 s.t. $z \in C_{n+1}$, and $v_1 \in C$. Let $\{x,y\} \subseteq W(G_1) \cap W(G_2)$ s.t. $xz \in E(G)$ but $yz \notin E(G)$. Let $v_2 \in W(G_2)-W(G_1)$. Let $x_1 \in C_1-C$. Let $P= z,v_n,v_{n-1},\ldots,v_2$ be a shortest z-v_2 path in $G_2-(C-v_2)$. If P is of length one, then $G[\{x,y,x_1,v_1,v_2,v_3,z\}]$ is isomorphic to H_5', where $v_3 \in (C_n \cap C_{n+1})-C$. Next assume that P is of length at least two. Then $G[\{v_1,v_2,\ldots,v_n,x,y,x_1,z\}]$ is isomorphic H_4'.∎

Now $\mathcal{A}(G,C)$, $G \in \mathcal{F}_{\mathcal{C}_1}$, is isomorphic to an induced odd cycle C_{2k+1}, $k \ge 1$. Next we classify the odd cycle of $\mathcal{A}(G,C)$ into three types and tackle each type separately. To this end the indices are under modulo 2k+1.

Let $\alpha = G_1, G_2, \ldots, G_{2k+1}$, $k \ge 1$, be the induced odd cycle of $\mathcal{A}(G,C)$. α is said to be of 1st type if there exists i s.t. $W(G_i) = W(G_{i+1})$. α is said to be of 2nd type if (1) it is not of 1st type and (2) there exists i s.t. either $W(G_{i+1})$ or $W(G_{i-1})$ is properly contained in $W(G_i)$. If α is neither, it is of 3rd type.

A separating clique C of G is said to be strong if G has maximum number of separated graphs w.r.t. C.

We next define some graphs which will appear in the list of forbidden subgraphs for DV graphs. We define $A_{11}(k>1)$, $A_{12}(k>1)$, $A_{13}(k>1)$, and $A_{14}(k>1)$ as follows:

$V(A_{11}(k>1))= \{v_1,v_2,\ldots,v_{2k+1},u_1,u_2,\ldots,u_{2k+1},x_1\}$, $k>1$ s.t. $\{v_1,v_2,\ldots,v_{2k+1}\}$ is a K_{2k+1}, u_i is joined to v_{i-1} and v_i, $2\le i\le 2k+1$, u_1 is joined to each of $v_1,v_2,\ldots,v_{2k},x_1$, and x_1 is joined to u_1 and v_1.

$A_{12}(k>1)$ is the trampoline T_{2k+1}.(For definition of trampoline see Chapter 1)

$V(A_{13}(k>1))= \{v_1,v_2,\ldots,v_{2k+1},u_1,u_2,\ldots,u_{2k+1}\}$, $k>1$ s.t. $\{v_1,v_2,\ldots,v_{2k+1}\}$ is a K_{2k+1}, u_i is joined to v_i and v_{i+1}, $2\le i\le 2k+1$, and u_1 is joined to each of $v_2,\ldots,v_{2k}$.

$V(A_{14}(k>1))= \{v_1,v_2,\ldots,v_{2k},u_1,u_2,\ldots,u_{2k+1}\}$, $k>1$ s.t. $\{v_1,v_2,\ldots,v_{2k}\}$ is a K_{2k}, u_i is joined to v_{i-1} and v_i, $2\le i\le 2k$, u_1 is joined to each of $v_2,\ldots,v_{2k}$, and u_{2k+1} is joined to each of $v_1,v_2,\ldots v_{2k-1}$.

The graphs $A_{11}(2)$, $A_{12}(2)$, $A_{13}(2)$, and $A_{14}(2)$ are given in Figure 2.3.2.

Theorem 2.3.6: $G \in \mathcal{F}_{\mathcal{C}_1}$ iff G is isomorphic to either one of the graphs in Figure 2.3.3 or one of the graphs $A_{11}(k>1)$ to $A_{14}(k>1)$.

Proof: Sufficiency:

It is easy to check that each of the graphs in Figure 2.3.3 and each of the graphs $A_{11}(k>1)$ to $A_{14}(k>1)$ is a critical DV graph.

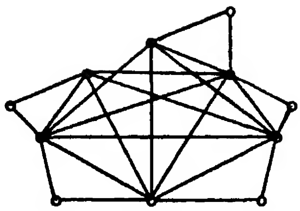
Necessity:

If G is not chordal, then it must be isomorphic to C_n, $n\ge 4$, which is A_{15} of Figure 2.3.3. Let C be a strong separating clique of G. Since $G \in \mathcal{F}_{\mathcal{C}_1}$, by Lemma 2.3.2, $\mathcal{A}(G,C)$ is isomorphic to an induced odd cycle $\alpha = G_1, G_2,\ldots, G_{2k+1}$.

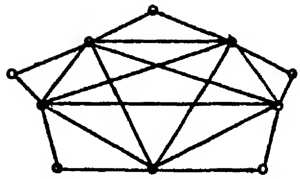
Case 1: $k=1$.

Subcase 1(a): α is of 1st type.

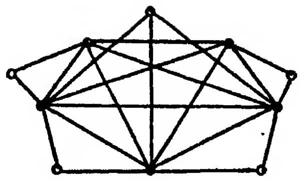
Wlg, $W(G_1)=W(G_2)$. Let C_i be a principal clique of G_i, $1\le i\le 3$. So there exists a relevant nonprincipal clique C_i' in G_i, $i=1,2$. If $W(G_3) \subseteq W(G_2)$, then take $C'=C_2$, and if $W(G_2) \subseteq W(G_3)$, then take $C'=C_3$. So if $W(G_2)$ and



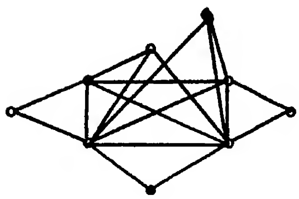
$A_{11}(2)$



$A_{12}(2)$



$A_{13}(2)$



$A_{14}(2)$

Figure 2.3.2: Illustration of Some Critical DV-Graphs.

$W(G_3)$ are comparable, then $\mathcal{A}(G,C')$ has at least four points, and hence the choice of C is contradicted. So $C_3 \Leftrightarrow C_2$. Again we have seen in the proof of Lemma 2.3.5 subcase 2(a) that each of G_1 and G_2 has exactly two relevant cliques. Let $x_1 \in C_1'-C_1$, $x_2 \in (C_1' \cap C_1)-C$, $y_1 \in C_2'-C_2$, $y_2 \in (C_2' \cap C_2)-C$, $z_1 \in C_3-C$, $z_2 \in W(G_3)-W(G_2)$.

Before proceeding further we prove the following claim.

Claim: C_1' is not antipodal to C_2'.

Proof of the Claim:

If possible, $C_1' \Leftrightarrow C_2'$. If $C_3>C_1'$ and $C_3>C_2'$, then let $z_3 \in W(G_2)-W(G_3)$, and $z_4 \in (C_1' \cap C_2' \cap C)$. Then $G[\{x_1,x_2,y_1,y_2,z_1,z_2,z_3,z_4\}]$ is isomorphic to A_1. But A_1 is a proper induced subgraph of G and is a critical DV graph. So we get a contradiction.

If $C_3|C_i'$, i=1,2, then let $z_3 \in W(G_2) \cap W(G_3)$, and $z_4 \in (C_1' \cap C_2' \cap C)$. Then $G[\{x_1,x_2,y_1,y_2,z_1,z_2,z_3,z_4\}]$ is isomorphic to A_2. Since A_2 is a proper induced subgraph of G and is a critical DV graph, we get a contradiction.

So either C_1' is attached to C_3 or C_2' is attached to C_3. Wlg, C_1' is attached to C_3. Let $G'=G-((C_2 \cap C_2')-C)$. Then $\alpha'=G_1',G_2',G_3'$ is an odd cycle of $\mathcal{A}(G',C_1)$, where G_1', G_2', and G_3' are the separated graphs of G' w.r.t. C_1 containing C_1',C_2', and C_3, respectively. So G' is not a DV graph. This contradicts the fact that G is a critical DV graph and proves our claim.

Suppose $(C_1' \cap C_2' \cap C_3) \neq \phi$. Now $W(G_2)-((C_1' \cup C_2') \cap C) \neq \phi$, as C_1' is not antipodal to C_2'. Wlg, $C_1' \geq C_2'$, and $|C_1' \cap C_2'|=1$. If $W(G_2)-((C_1' \cup C_2') \cap C) \subset W(G_3)$, then there exists $x' \in ((C_1' \cup C_2') \cap C)$ s.t. $x' \notin W(G_3)$. Let $G'=G-x'$. Then $\mathcal{A}(G',C_2)$ has an odd cycle $\alpha'=G_1',G_2',G_3'$, where G_i' is the separated graph of G' containing $G_i-W(G_i)$, $1 \leq i \leq 3$. So G is not a critical DV graph, which is a contradiction. So $W(G_2)-((C_1' \cup C_2') \cap C)$ is not a subset of $W(G_3)$. Let $x'' \in (C_1' \cap C_2' \cap C)$, and $z_3 \in (W(G_2)-((C_1' \cup C_2') \cap C_3))-W(G_3)$. Then $G[\{x_1,x_2,y_1,y_2,z_1,z_2,z_3,x''\}]$ is isomorphic to A_1.

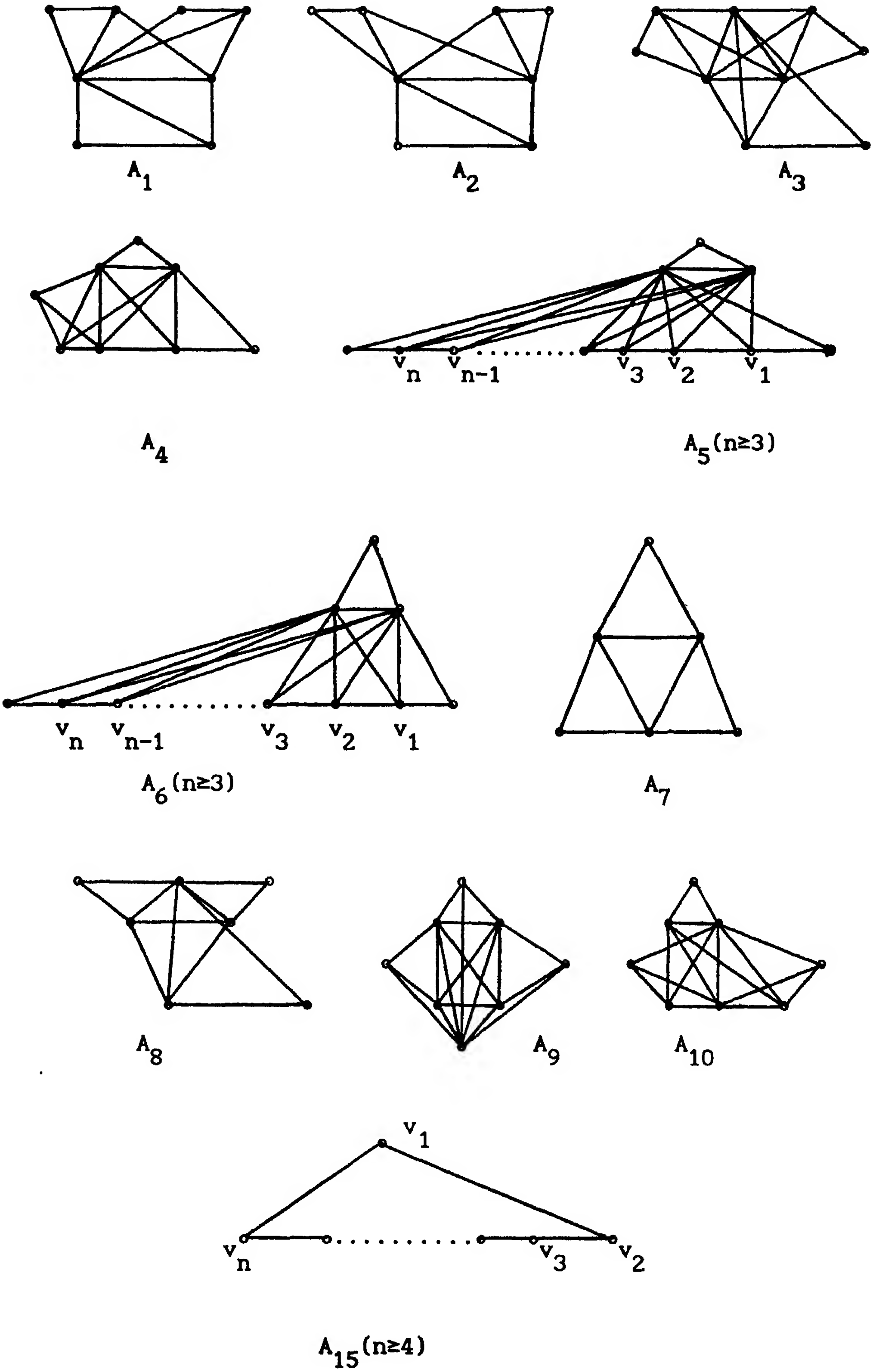


Figure 2.3.3: Forbidden Subgraphs For DV Graphs.

Next assume that $(C_1' \cap C_2' \cap C_3)=\phi$. If $(C_1' \cap C_2')\neq\phi$, then let $z_3 \in W(G_2)\cap W(G_3)$, and $z_4 \in C_1' \cap C_2'$. Then $G[\{x_1,x_2,y_1,y_2,z_1,z_2,z_3,z_4\}]$ is isomorphic to A_2. Assume that $C_1' \cap C_2'=\phi$. If $C_1'|C_3$ and $C_2'|C_3$, then let $z_3 \in C_1' \cap C$, $z_4 \in C_2' \cap C$, and $z_5 \in W(G_2)\cap W(G_3)$. Then $G[\{x_1,x_2,y_1,y_2,z_1,z_2,z_3,z_4,z_5\}]$ is isomorphic to A_3. Next wlg, assume that C_1' is attached to C_3. Let $z_3 \in C_1' \cap C_3$, and $z_4 \in C_2' \cap C$. Let $G'=G[\{x_1,x_2,y_1,y_2,z_1,z_2,z_3,z_4\}]$. If $z_4 \notin C_3$, then G' is isomorphic to A_2. If $z_4 \in C_3$, then let $z_5 \in W(G_2)-W(G_3)$. If either $z_5 \in C_1'$ or $z_5 \in C_2'$, then $G''= G[\{x_1,x_2,y_1,y_2,z_1,z_3,z_4,z_5\}]$ is isomorphic to A_4, as $C_1'|C_2'$. Now G'' is a proper induced subgraph of G, a contradiction. So neither $z_5 \in C_1'$ nor $z_5 \in C_2'$. Now $G''=G[\{x_1,x_2,y_1,y_2,z_1,z_3,z_4,z_5\}]$ is isomorphic to $A_6(n=3)$. Since G'' is a proper induced subgraph of G, we have a contradiction.

Subcase 1(b): α is of 2nd type.

Wlg, $W(G_2) \subset W(G_3)$. Let C_3 be a principal clique of G_3. Now C_3 is a separating clique of G. Now $W(G_1)$ is not a proper subset of $W(G_3)$, otherwise $\mathcal{A}(G,C_3)$ will have at least four points contradicting the choice of C. Let G_i' be the separated graphs of G w.r.t. C_3 containing $G_i-W(G_i)$, $1\le i\le 3$($W(G_i)$ is w.r.t. C). By the choice of C, there are exactly three separated graphs w.r.t. C_3. Let C_i' be a principal clique of G_i', $1\le i\le 3$. Then clearly $C_3' \Leftrightarrow C_1'$ and $C_1' > C_2'$, and either $C_2' \Leftrightarrow C_3'$ or $C_3' > C_2'$. Now by Lemma 2.3.5, $G_1' \cup G_2'$ contains a subgraph isomorphic to either H_4' or H_5'.

Assume that $G_1' \cup G_2'$ contains a subgraph isomorphic to $H_4'(n\ge3)$. Then either $C_2' \Leftrightarrow C_3'$ or, $G_2' \cup G_3'$ contains a subgraph isomorphic to H_4' , because if $G_2' \cup G_3'$ contains a subgraph isomorphic to H_5', then G will not be a critical DV graph, which can be seen from the structure of H_4' and H_5'. Suppose $G_2' \cup G_3'$ is isomorphic to $H_4'(n\ge3)$. If every relevant clique of G_3' is attached to every relevant clique of G_1', then G will be isomorphic to A_5; otherwise, G will be isomorphic to A_6. Next assume that $C_2' \Leftrightarrow C_3'$. Suppose $G_1' \cup G_2'$ is isomorphic to $H_4'(n>3)$. If C_3' is attached to every relevant clique of G_1',

then G will be isomorphic to A_5; otherwise, G will be isomorphic to A_6. If $G_1' \cup G_2'$ is isomorphic to $H_4'(n=3)$, then G will be isomorphic to one of A_5, A_6, and A_4.

Next, let $G_1' \cup G_2'$ contain a subgraph isomorphic to H_5'. Then clearly $C_2' \Leftrightarrow C_3'$, otherwise, $G_2' \cup G_3'$ will contain a subgraph isomorphic to H_4' or H_5', in which case it can be seen using the structure of H_4' and H_5' that G is not a critical DV graph. So G will be isomorphic to A_4 or A_{10} in this case. (Note that if $G_2' \cup G_3'$ is also isomorphic to H_5', then G will contain a subgraph isomorphic to the graph F_1 in Figure 2.4.3, which is a critical UV graph but not a critical DV graph. We will use this in the next section in finding the forbidden subgraphs for UV graphs.)

Subcase 1(c): α is of 3rd type.

If $W(G_1) \cap W(G_2) \cap W(G_3) = \emptyset$, then let $x \in W(G_1) \cap W(G_2)$, $y \in W(G_2) \cap W(G_3)$, and $z \in W(G_3) \cap W(G_1)$, $x_i \in C_i - W(G_i)$, where C_i is a principal clique of G_i, $1 \le i \le 3$. Then $G[\{x,y,z,x_1,x_2,x_3\}]$ is isomorphic to A_7. Otherwise let $x \in W(G_1) \cap W(G_2) \cap W(G_3)$. Again $C'=C-x$ is a separating clique of $G'=G-x$. Let $G_i'=G_i-x$. Then G_1', G_2', and G_3' are the only separated graphs of G' w.r.t. C'. Note that $G_i' \Leftrightarrow G_j'$ iff G_i' is attached to G_j'. If G_1', G_2' and G_3' are pair wise unattached, then let $x_i' \in W(G_i')$ and $y_i \in C_i' - W(G_i')$, where C_i' is a principal clique of G_i', $1 \le i \le 3$. Then $G[\{x,x_1',x_2',x_3',y_1,y_2,y_3\}]$ is isomorphic to A_8. If $\mathcal{A}(G',C')$ is connected, wlg, assume that $G_1' \Leftrightarrow G_2'$, $G_2' \Leftrightarrow G_3'$, and $G_1' | G_3'$. Let $x_1 \in W(G_1') \cap W(G_2')$, $x_2 \in W(G_2') \cap W(G_3')$, $x_3 \in (C_1' \cap C') - C_2'$, $x_4 \in (C_3' \cap C') - C_2'$, where C_i' is a principal clique of G_i', $1 \le i \le 3$. Then $G[\{x,x_1,x_2,x_3, x_4,y_1,y_2,y_3\}]$ is isomorphic to A_9. So assume that $G_1' \Leftrightarrow G_2'$, $G_3' | G_1'$, and $G_3' | G_2'$. Let $x_1 \in W(G_1') - W(G_2')$, $x_2 \in W(G_2') - W(G_1')$, $x_3 \in W(G_3')$, $y_i \in C_i' - C$, where C_i' is a principal clique of G_i', $1 \le i \le 3$. Then $G[\{x,x_1,x_2,x_3,y_1,y_2,y_3\}]$ is isomorphic to A_8, which is a proper induced subgraph of G. Hence we have a contradiction.

Case 2: $k > 1$.

Subcase 2(a): α is of 1st type.

Wlg, $W(G_1) = W(G_2)$. Since $k>1$, G_1 and G_2 are strong separated graphs. So $W(G_i) \subseteq W(G_1)$ for all i, $2 \le i \le 2k+1$. Since $G_1 \Leftrightarrow G_{2k+1}$, and $W(G_{2k+1}) \subset W(G_1)$, G_1 has at least two relevant cliques. Let C_1 be a principal clique of G_1. Then clearly C_1 is a separating clique of G, and $\mathcal{A}(G,C_1)$ has more vertices than $\mathcal{A}(G,C)$. This contradicts the choice of C. Hence α can not be of 1st type.

Subcase 2(b): α is of 2nd type.

Wlg, $W(G_2) \subset W(G_1)$. We claim that G_1 dominates G_j, $3 \le j \le 2k$. Now $G_3 \Leftrightarrow G_2$. As $k>1$, $W(G_3)$ and $W(G_1)$ are comparable. If $W(G_1) \subseteq W(G_3)$, then since $G_2 \Leftrightarrow G_3$, $G_1 \Leftrightarrow G_3$, and $k>1$, it contradicts the fact that α is chordless. So G_1 dominates G_3 and hence G_1 dominates G_j, $3 \le j \le 2k$, by Lemma 2.3.1. Again by Lemma 2.3.1, $W(G_i)$ is incomparable with $W(G_{i+1})$, $2 \le i \le 2k-1$. Let C_i be a principal clique of G_i, $1 \le i \le 2k+1$. Now $G_1 \Leftrightarrow G_2$, and $W(G_2) \subset W(G_1)$. So there exists C_1' in G_1 s.t. $C_2 > C_1'$. Note that $W(G_{2k+1})$ is not a subset of $W(G_1)$; otherwise, C_1 will be a separating clique of G s.t. $\mathcal{A}(G,C_1)$ has more vertices than that of $\mathcal{A}(G,C)$ contradicting the choice of C. So $W(G_1)$ and $W(G_{2k+1})$ are incomparable. Assume that G_{2k+1} dominates some G_i, $i \notin \{1,2k\}$. Then G_{2k+1} is a strong separated graph. Let $G'=G-(C-W(G_1))$. Then C_1 is a separating clique of G' and $\mathcal{A}(G',C_1)$ has an induced odd cycle, namely $G_1', G_2', \ldots, G_{2k+1}'$, where G_i' is the separated graph of G' w.r.t. C_1 containing C_i, $2 \le i \le 2k+1$, and G_1' is the separated graph containing C_1'. So G' is not a DV graph, which is contrary to the fact that $G \in \mathcal{F}_{\mathcal{E}_1}$. So G_{2k+1} does not dominate any G_i, $1 \le i \le 2k$. So $W(G_{2k})$ is not a subset of $W(G_{2k+1})$. Hence $C_i \Leftrightarrow C_{i+1}$, $2 \le i \le 2k$. Let $x_i \in (C_i \cap C_{i+1})$, $y_i \in C_i - C$, $2 \le i \le 2k$, $x_1 \in (C_1 \cap C_1' \cap C)$, $x_{2k+1} \in (C_{2k+1} \cap C) - C_1$, $y_1 \in (C_1' \cap C_1) - C$, and $y_1' \in C_1' - C_1$. Then $G[\{x_1, x_2, \ldots, x_{2k+1}, y_1, y_2, \ldots, y_{2k+1}, y_1'\}]$ is isomorphic $A_{11}(k>1)$.

Case 3: α is of 3rd type.

Subcase 3(a): $G_i | G_j$ iff $i \neq j-1$, $1 \leq i < j \leq 2k+1$.

Let C_i be a principal clique of G_i, $1 \leq i \leq 2k+1$. Let $x_i \in C_i \cap C_{i+1}$, and $y_i \in C_i - C$, $1 \leq i \leq 2k+1$. Then $G[\{x_1, x_2, \ldots, x_{2k+1}, y_1, y_2, \ldots, y_{2k+1}\}]$ is isomorphic to A_{12}.

Subcase 3(b): There exists some G_i which dominates some separated graphs.

Now by Lemma 2.3.1, G_i is a strong separated graph. If G_i is the only strong separated graph, then using the analysis used earlier it can be shown that G is isomorphic to A_{13}, as α is of 3rd type. If there exists some other strong separated graph G_j, then by Lemma 2.3.1, G_i and G_j are the only strong separated graphs, and G_i and G_j occur consecutively in α. Using a similar analysis it can be shown that G will be isomorphic to A_{14} in this case.∎

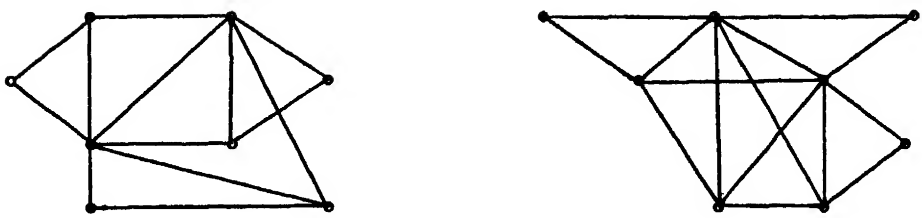
2.4 Forbidden Subgraphs For UV Graphs:

The concept of Path (also UV) graphs was introduced by Renz[108]. In [108] he presented two minimal forbidden subgraphs R_1, and R_2 of Figure 2.4.1, for Path graphs, and asked to find all other forbidden subgraphs for Path graphs. In this section we study the structure of forbidden subgraphs for Path graphs.

Throughout this section, let $G \in \mathcal{F}_{\mathcal{C}_2}$ and C be any separating clique of G. We have the following useful observations.

Observation 2.4.1 (i) $\mathcal{A}(G,C)$ has an induced odd cycle $\alpha = G_1, G_2, \ldots, G_{2k+1}$.
(ii) Every clique of G except C is a relevant clique of G w.r.t.C.

G is said to be of 1st type if $\mathcal{A}(G,C)$ has an induced odd cycle of 1st type. G is of 2nd type if G is not of 1st type and $\mathcal{A}(G,C)$ has an induced odd cycle of 2nd type. If G is neither of 1st type nor of 2nd type, then it is said to be of 3rd type. A triplet (G_1, G_2, G_3) is said to be an



R_1 R_2

Figure 2.4.1: The two Forbidden Subgraphs for Path Graphs Due to Renz.

antipodal triplet w.r.t. some $v \in V$ if they are pair wise antipodal and $v \in (W(G_1) \cap W(G_2) \cap W(G_3))$. In this case we denote it by (v,G_1,G_2,G_3).

Lemma 2.4.2: Let $G \in \mathcal{F}_{C_2}$ be of 1st type w.r.t C. Then either (i) G has an antipodal triplet, or (ii) $\mathcal{A}(G,C)$ contains an induced odd cycle of length at least 5 containing two consecutive strong separated graphs.

Proof: Let $\alpha = G_1, G_2,..G_i,..., G_{2k+1}$, $k \geq 1$ be an induced odd cycle of 1st type of $\mathcal{A}(G,C)$. Assume that G has no antipodal triplet. Wlg, we may take $W(G_1)=W(G_2)$, whence k>1. If $W(G_3)$ is not a proper subset of $W(G_2)$, then $G_1 \Leftrightarrow G_3$ because $G_2 \Leftrightarrow G_3$. This is contrary to the fact that α is chordless. So $W(G_3) \subset W(G_2)$ and G_1 dominates G_3. Hence by Lemma 2.3.1, G_1 is a strong separated graph. Similarly G_2 is a strong separated graph. Now by Lemma 2.3.1 G_1 and G_2 are the only strong separated graphs in α.∎

Lemma 2.4.3: Let $G \in \mathcal{F}_{\mathcal{C}_2}$ be of 2nd type w.r.t. C. Then $\mathcal{A}(G,C)$ is isomorphic to an induced odd cycle $\alpha = G_1, G_2,...,G_{2k+1}$. Moreover, either (i) α contains a strong separated graph, or (ii) G has an antipodal triplet (x, G_1, G_2, G_3).

Proof: Let $\alpha = G_1, G_2,...,G_{2k+1}$ be an induced odd cycle of 2nd type. Wlg, $W(G_2) \subset W(G_1)$. If $W(G_3)$ is not a subset of $W(G_1)$, then $G_3 \Leftrightarrow G_1$. Let $x \in W(G_2) \cap W(G_3)$. Then (x, G_1, G_2, G_3) is an antipodal triplet. So $W(G_3) \subseteq W(G_1)$. If k=1, then again G has an antipodal triplet. If k >1, then G_1 dominates G_3. So by Lemma 2.3.1, G_1 is a strong separated graph. Therefore, $W(G_i) \cap W(G_{i+1}) \subseteq W(G_1)$, $1 \leq i \leq 2k$. Hence $\bigcup_{i=1}^{2k+1} G_i$ is not a path graph. As $G \in \mathcal{F}_{\mathcal{C}_2}$, $\mathcal{A}(G,C)$ isomorphic to α.∎

Let G_i and G_j be any two separated graphs of a chordal graph G w.r.t. some separating clique. G_i is said to predominate G_j if G_i is attached to G_j, and $W(G_j) \subseteq W(G_i)$.

Lemma 2.4.4: Let $G \in \mathcal{F}_{\mathcal{C}_2}$ be of 3rd type w.r.t. some separating clique C and

G has no antipodal triplet w.r.t. any separating clique. Let $\alpha=G_1,G_2,\ldots,G_{2k+1}$, $k>1$ be an induced odd cycle of G. Then either (1) α contains at least one strong separated graph, or (2) there exists a separated graph G′ s.t. G′ predominates at least 2k separated graphs of α.

Proof: Assume that (1) is not true. Since $G \in \mathcal{F}_{\mathcal{C}_2}$, some $x \in C$ belongs to three differently colored separated graphs, say G′, G", and G"′, in any proper coloring of $\mathcal{A}(G,C)$. Since G does not contain any antipodal triplet w.r.t. any separating clique, G′,G", and G"′ are not pair wise antipodal. So wlg, $G' \geq G"$. We claim that G′ is different from G_i, $1\leq i\leq 2k+1$. If possible, wlg, $G'=G_1$. Let C_i be a principal clique of G_i, $1\leq i\leq 2k+1$. So C_1 is a separating clique of G. But $C_4 | C_1$, which contradicts our observation 2.4.1(ii). So G′ is different from G_i, $1\leq i\leq 2k+1$. To complete the proof of the Lemma, we need the following intermediate results.

Fact 1: G′ predominates some separated graphs of α.

Proof of Fact 1: If possible, let G′ do not predominate any of G_i, $1\leq i\leq 2k+1$, where $\alpha = G_1, G_2,\ldots, G_{2K+1}$. Let C′ be a principal clique of G′. Then clearly C′ is a separating clique of G, as G′ dominates some separated graph. Since α is of 3rd type there exists C_i in G_i s.t. $C_i \Leftrightarrow C_{i+1}$, $1\leq i\leq 2k+1$. Let $x \in W(G_1)\cap W(G_2)$, and G^1 be the separated graph w.r.t. C′ containing $V(G_1)-W(G_1)$, where G_1 and $W(G_1)$ are w.r.t. C. Then G^1 contains $V(G_i)-W(G_i)$, where G_i and $W(G_i)$ are w.r.t. C, $1\leq i\leq 2k+1$. Let $G'_1, G'_2,\ldots,G'_r$ be the separated graphs of G w.r.t. C′ s.t. $G'_r= G^1$. Now C′ is again a separating clique of G-x, and $G'_1, G'_2,\ldots, G'_{r-1}$ are also separated graphs of G-x w.r.t. C′. Let $G'_r(x)= G'_r-x$. Clearly $G'_i \cap G'_j \cap G'_r \neq \phi$ iff $G'_i \cap G'_j \cap G'_r(x) \neq \phi$, $1\leq i<j<r$. Next we claim that $G'_i\Leftrightarrow G'_r$ iff $G'_i\Leftrightarrow G'_r(x)$, $1\leq i<r$. Clearly $G'_i\Leftrightarrow G'_r(x)$ implies $G'_i\Leftrightarrow G'_r$. Suppose $G'_i\Leftrightarrow G'_r$, and G'_i is not antipodal to $G'_r(x)$. So $G'_r(x) > G'_i$. Let C'_i and C'_r-x be some principal cliques of G'_i and $G'_r(x)$, respectively. Since $G'_i\Leftrightarrow G'_r$, there exist some C''_r in G'_r s.t. $C'_i > C''_r$, and $x \in$

C''_r. So $C_1 > C''_r$ or $C_2 > C''_r$. Wlg, $C_1 > C''_r$. Then C_1 is a separating clique of G but $C_4|C_1$, which is a contradiction to the observation 2.4.1(ii). So $G'_i \leftrightarrow G'_r$ iff $G'_i \leftrightarrow G'_r(x)$. Hence G-x is not a path graph as G is not a path graph. This contradicts the fact that $G \in \mathcal{F}_{\mathcal{C}_2}$, and proves the Fact 1.∎

Assume that G′ predominates at most (2k-1) separated graphs of α.

Fact 2: G′ does not predominate some two consecutive separated graphs of α.

Proof of Fact 2: Wlg G′ does not predominate G_1. Let $x \in W(G_1)-W(G')$. Since α is of 3rd type, there exists C_i in G_i s.t. $C_i \Leftrightarrow C_{i+1}$, $1 \le i \le 2k+1$. If possible, for every i, $1 \le i \le 2k+1$, let G_i or G_{i+1} be predominated by G′. Then $|\ W(G_i) \cap W(G')| \ge 2$, for all i= 1,2,...,2k+1. Let $\{ w'_i, w''_i\} \subseteq (W(G_i) \cap W(G'))$ s.t. $w'_i \in (W(G_i) \cap W(G_{i-1}))$ and $w''_i \in (W(G_i) \cap W(G_{i+1}))$. Let $w'''_i \in C_i - C$, where C_i is a principal clique of G_i, $1 \le i \le 2k+1$. Let C′ be a principal clique of G′.Then $G[\{w'_i, w''_i, w'''_i\}_{i=1}^{2k+1} \cup C' \cup \{x\}]$ is a proper subgraph of G, which is not a path graph. Hence we have a contradiction and the Fact 2 is established.∎

Fact 3: G′ predominates at least 2k-1 separated graphs of α.

Proof of fact 3: Assume that G′ predominates at most 2k-2 separated graphs. So by Fact 2, G′ does not predominate some two consecutive separated graphs of α. Wlg, G′ does not predominate G_1, G_2, and G_i. Let $x \in W(G_1) \cap W(G_2)$, and $y \in W(G_i)-W(G')$. Let G'_i, $1 \le i \le r$, be the separated graph of G w.r.t. C′, where C′ is a principal clique of G′ s.t. $y \in G'_r$. So as in the proof of the Fact 1, G-x will not be a path graph, which is a contradiction. So G′ predominates at least 2k-1 separated graphs of α.∎

Suppose G′ does not predominate G_1 and G_2 but predominates G_3. Let C′ be a principal clique of G′. Then clearly C′ is a separating clique of G. Let S be the set of separated graphs w.r.t. C predominated by G′. Let $G^* = G_{i_r}$, where P= G_3, $G_{i_1}, G_{i_2}, \ldots, G_{i_{r-1}}, G_{i_r}$ is a longest path of $\mathcal{A}(G,C)$ starting from G_3 among all the paths of $\mathcal{A}(G,C)$ containing only the

separated graphs from S. Clearly length of P is at least 2k-2. Now corresponding to each member G_i of S there is a separated graph $G_i(1)$ of G w.r.t. C′ containing $G_i-(C-W(G_i)$($W(G_i)$ is w.r.t. C). Let $S(1)=\{G_i(1)$ s.t. $G_i \in S\}$. Let G^{**} be the separated graph of G w.r.t. C′ containing $G_1-W(G_1)$ ($W(G_1)$ is w.r.t. C).Let S(2)={ G_i' s.t. G_i' is a separated graph w.r.t. C′ and neither $G_i' \in S(1)$ nor $G_i'=G^{**}$ }.

Since G is a critical path graph, H $=(G-G^{*}(1))\cup C'$ is a path graph. Let f be a coloring function of $\mathcal{A}(H,C')$. Since $W(G_{i_j}(1)) \le W(G^{**})$, $1\le j\le r$, and $W(G_3(1)) \le W(G^{**})$, the separated graphs of the path $P'= G_3(1),G_{i_1}(1)$, $G_{i_2}(1),\ldots,G_{i_{r-1}}(1)$,receives exactly two colors, say 1 and 2, under f. Wlg, $G_{i_{r-1}}(1)$ receives color 1 under f. Color $G^{*}(1)$ by color 2.

Claim A: The new coloring is a valid coloring in the sense that it does not violate any condition of the separator Theorem for UV graphs.

First we show that the coloring of $\mathcal{A}(G,C')$ is a proper coloring. If possible, there exists G_i' s.t. $G_i' \Leftrightarrow G^{*}(1)$, and G_i' has color 2. By the choice of $G^{*}(1)$, $G_i' \notin S(1)$. If $G_i'=G^{**}$, then $P'\cup\{G^{**}G_3(1),\ G^{**}G^{*}(1)\}$ is an odd cycle s.t. $W(G_{i_j}(1) \le W(G^{**})$, $1\le j\le r$, where $G^{*}(1)=G_{i_r}(1)$. So $\mathcal{A}(G,C')$ will have an induced odd cycle α containing a strong separated graph, which contradicts our assumption. Next assume that $G_i' \in S(2)$. So G^{**} has color 1. Hence $P'\cup\{G^{**}G_3(1),\ G^{**}G^{*}(1)\}$ is an odd cycle s.t. $W(G_{i_j}(1) \le W(G^{**})$, $1\le j\le r$, where $G^{*}(1)=G_{i_r}(1)$. So $\mathcal{A}(G,C')$ will have an induced odd cycle α containing a strong separated graph, which contradicts our assumption. Hence the new coloring is a proper coloring of $\mathcal{A}(G,C')$.

Next we show that there is no $x \in C'$ s.t. x has three differently colored neighboring separated graphs. If possible, let there exist G_1', G_1'' s.t. G_1', G_1'', and $G^{*}(1)$ receive different colors, and $W(G_1')\cap W(G_1'')\cap W(G^{*}(1)$

$\neq\phi$. Let $x \in W(G_1')\cap W(G_1'')\cap W(G^*(1))$.

Case 1: G_1' and G_1'' both belong to $S(2)$.

Then G^{**}, G_1', and G_1'' are three differently colored separated graphs s.t. $x \in W(G_1')\cap W(G_1'')\cap W(G^{**})$. This contradicts the fact that f is a valid coloring of $\mathcal{A}(G,C')-G^*(1)$.

Case 2: One of G_1', and G_1'' belongs to $S(2)$.

Wlg, $G_1' \in S(2)$. Now $G_1'' \in S(1)$ or $G_1''=G^{**}$. First assume that $G_1''=G^{**}$. Then clearly G^{**} has color 1. If $G_1' \Leftrightarrow G^*(1)$, then $P'\cup\{G^{**}G_3(1), G_1'G^*(1), G_1'G^{**}\}$ is an odd cycle s.t. $W(G_{1_j}(1) \subseteq W(G^{**})$, $1\leq j\leq r$, where $G^*(1)=G_{1_r}(1)$, and $W(G^*(1) \cap W(G_1') \subseteq W(G^{**})$. So $\mathcal{A}(G,C')$ will have an induced odd cycle α containing a strong separated graph, a contradiction to our assumption. If $G_1'\geq G^*(1)$, then $G_{1_{r-2}}(1)$ is attached to G_1'. So $G^{**}, G_{1_{r-2}}(1)$, and G_1' are differently colored separated graphs s.t. $W(G^{**})\cap W(G_{1_j}(1))\cap W(G_1') \neq\phi$. This contradicts the fact that f is a valid coloring of $\mathcal{A}(G,C')-G^*(1)$.

Next assume that $G_1'' \neq G^{**}$. If G^{**} has color 2, then G^{**}, G_1', and G_1'' are differently colored separated graphs s.t. $W(G^{**})\cap W(G_1'')\cap W(G_1') \neq\phi$. This contradicts the fact that f is a valid coloring of $\mathcal{A}(G,C')-G^*(1)$. So assume that G^{**} has color 1. If $G_1' \Leftrightarrow G^*(1)$, then $P'\cup\{G^{**}G_3(1), G_1'G^*(1), G_1'G^{**}\}$ is an odd cycle s.t. $W(G_{1_j}(1) \subseteq W(G^{**})$, $1\leq j\leq r$, where $G^*(1)=G_{1_r}(1)$, and $W(G^*(1) \cap W(G_1') \subseteq W(G^{**})$. So $\mathcal{A}(G,C')$ will have an induced odd cycle α containing a strong separated graph, which is a contradiction to our assumption. So assume that $G_1'\geq G^*(1)$. Then $G_{1_{r-2}}(1)$ is attached to G_1'. So $G^{**}, G_{1_{r-2}}(1)$, and G_1' are differently colored separated graphs s.t. $W(G^{**})\cap W(G_{1_j}(1))\cap W(G_1') \neq\phi$. This contradicts the fact that f is a valid coloring of $\mathcal{A}(G,C')-G^*(1)$. So case 2 will not occur.

Case 3: Neither G_1' nor G_1'' belong to $S(2)$.

If neither $G_1'=G^{**}$ nor $G_1''=G^{**}$, then clearly G^{**} has color 1. Wlg, G_1'

has color different from 1. So we can replace G''_1 by G^{**}. So wlg, assume that $G''_1=G^{**}$. By the choice of $G^*(1)$, G'_1 is not antipodal to $G^*(1)$. First assume that $G'_1 \geq G^*(1)$. Then $G_{i_{r-2}}(1)$ is attached to G'_1. So G^{**}, $G_{i_{r-2}}(1)$, and G'_1 are differently colored separated graphs s.t. $W(G^{**}) \cap W(G_{i_j}(1)) \cap W(G'_1) \neq \phi$. This contradicts the fact that f is a valid coloring of $\mathcal{A}(G,C')-G^*(1)$. Next assume that $G^*(1) \geq G'_1$. So $C^*(1)$ is a separating clique of G, where $C^*(1)$ is a principal clique of $G^*(1)$. Again there exist G'_2 and G'_3 in S(2) s.t. $G'_2 \leftrightarrow G'_3$. By Observation 2.4.1(ii) both G'_2, and G'_3 are attached to $G^*(1)$. Since G has no antipodal triplet w.r.t. any separating clique, either $G'_2 \geq G^*(1)$ or $G'_3 \geq G^*(1)$. Wlg, $G'_2 \geq G^*(1)$. Let $x \in W(G'_2) \cap W(G'_3) \cap W(G^*(1))$. So (x, G^{**}, G'_2, G'_3) is an antipodal triplet of G as $W(G^*_1) \subseteq W(G^{**})$. Hence we have a contradiction. So case 3 will not occur.

Thus the new coloring is a valid coloring of $\mathcal{A}(G,C')$ and claim A is proved.

Since $\mathcal{A}(G,C)$ admits a valid coloring, G is not a critical path graph, a contradiction! Hence G′ predominates at least 2k separated graphs of α.∎

$G \in \mathcal{F}_{\mathcal{C}_2}$ is said to be a 'bad minimal separated graph' if (i) G is of 3rd type, (ii) every induced odd cycle of $\mathcal{A}(G,C)$ is of length 3 for every separating clique C of G, and (iii) there exists no odd cycle say $\alpha=G_1,G_2,G_3$ and a separated graph G′ w.r.t. C s.t. G′ predominates at least two separated graphs of α.

We next define two graphs $F_3(k)$, and $F_4(k)$ which will appear in the list of forbidden subgraphs for UV graphs.

$V(F_3(k))=\{v_1,v_2,\ldots,v_{2k+1},u_1,u_2,\ldots,u_{2k+1},x_1,x_2\}$, s.t. (1) $\{v_1,v_2,\ldots,v_{2k+1},u_1,u_2,\ldots,u_{2k+1}\}$ is a trampoline T_{2k+1} with $\{v_1,v_2,\ldots,v_{2k+1}\}$ is K_{2k+1}, and (2) x_i is adjacent to all of $\{v_1,v_2,\ldots,v_{2k+1}\}$, i=1,2.

The graph $F_4(k)$ is obtained from $F_3(k)$ by joining the vertices x_1 and u_1 in $F_3(k)$.

The graphs $F_3(2)$ and $F_4(2)$ are illustrated in Figure 2.4.2.

Theorem 2.4.5: $G \in \mathcal{F}_{\mathcal{E}_2}$ iff either G is a bad minimal forbidden subgraph or G is isomorphic to one of the graphs in A_1 to A_{15} mentioned in Theorem 2.3.6 except A_7 and A_{12} or a graph in Figure 2.4.3, or one of $F_3(k)$ and $F_4(k)$.

Proof: Sufficiency:

It is a routine job to verify that each of the graphs mentioned in Theorem 2.4.5 is a critical path graph.

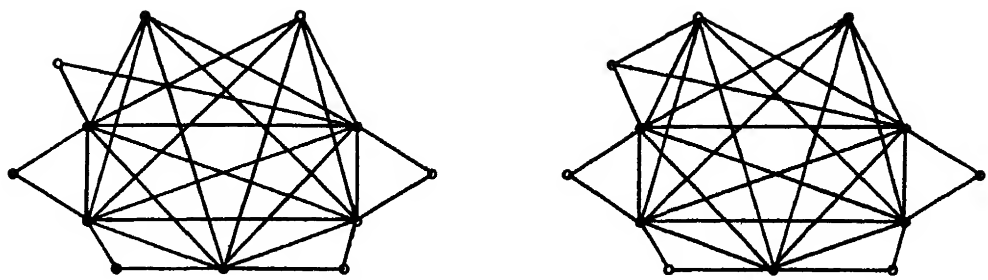
Necessity:

Let $G \in \mathcal{F}_{\mathcal{E}_2}$. If G is not chordal, then G will be isomorphic to C_n, $n \geq 4$. So assume that G is chordal. As G has more than two cliques, G has a separating clique. Let C be a strong separating clique of G, and let G_i, $1 \leq i \leq r$, be the separated graphs.

Case 1: G contains an antipodal triplet (x, G_1, G_2, G_3).

Let $\alpha = G_1, G_2, G_3$. If G is a critical DV graph, then as in Theorem 2.3.6, G will be isomorphic to one of A_1 to A_{10} except A_7 in Figure 2.3.2. So assume that G is not a critical DV graph. If α is not of 3rd type, then as in the proof of Theorem 2.3.6, G will be isomorphic to F_1. Next assume that α is of 3rd type. Let $G' = G - x$, and $G_i' = G_i - x$, $1 \leq i \leq 3$, and $C' = C - x$. Then C' is a separating clique of G' and G_1', G_2', and G_3' are the only separated subgraphs w.r.t. C'. Let C_i' be a principal clique of G_i', $1 \leq i \leq 3$.

If G_i's are pair wise antipodal, then as in Theorem 2.3.6, G' will be isomorphic to A_7, since G' is also of 3rd type. So $G[V(A_7) \cup \{x\})]$ will be isomorphic to F_2 of Figure 2.4.3.. Since G is a critical path graph, G will be isomorphic to F_2.



$F_3(2)$ $F_4(2)$

Figure 2.4.2: Two critical UV-Graphs

If $\mathcal{A}(G',C')$ is disconnected, then assume wlg that G_1' and G_2' lie in different components of $\mathcal{A}(G',C')$. Let $x_1 \in W(G_1')$, $x_2 \in W(G_2')-W(G_3')$, $x_3 \in W(G_3')-W(G_2')$, $y_i \in C_i'-C'$, $1 \le i \le 3$. Then $G[\{x,x_1,x_2,x_3,y_1,y_2,y_3\}]$ is isomorphic to A_8 of Figure 2.3.3, which is a critical DV graph. This contradicts our assumption.

If $G_1' \leftrightarrow G_2'$, $G_2' \Leftrightarrow G_3'$, and $G_1' \mid G_3'$. Then let $x_1 \in W(G_1')-W(G_2')$, $x_2 \in W(G_2') \cap W(G_1')$, $x_3 \in W(G_2') \cap W(G_3')$, $x_4 \in W(G_3')-W(G_2')$, and $y_i \in C_i-C'$, $1 \le i \le 3$. Then $G[\{x,x_1,x_2,x_3,x_4,y_1,y_2,y_3\}]$ is isomorphic to A_9 of Figure 2.3.3, which is a critical DV graph. This contradicts our assumption.

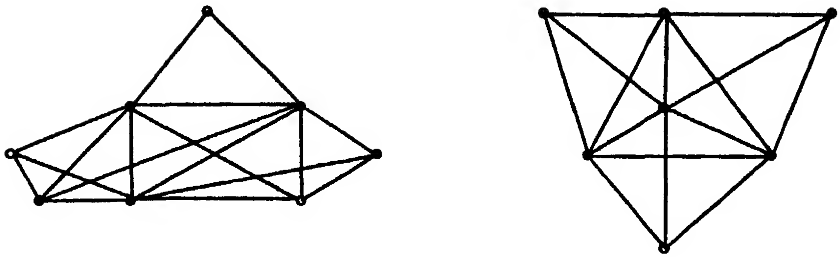
Case 2: G does not have any antipodal triplet with respect to any separating clique.

Subcase 2(a): G is of 1st type.

Since G does not have any antipodal triplet, by Lemma 2.4.2, $\mathcal{A}(G,C)$ contains an odd cycle containing two strong separated graphs. Let $\alpha = G_1, G_2, \ldots, G_{2k+1}$. Wlg, $W(G_1)=W(G_2)$. So G_1 and G_2 are the two strong separated graphs. Hence $W(G_i) \subseteq W(G_1)$ for all i, $2 \le i \le 2k+1$. Since $G_1 \Leftrightarrow G_2$, by Lemma 2.3.5, each of G_1 and G_2 has at least two relevant cliques. Let C_1 be a principal clique of G_1. Then C_1 is separating clique of G and $\mathcal{A}(G,C_1)$ has more vertices than that of $\mathcal{A}(G,C)$. This contradicts the choice of C. So this case will not occur.

Subcase 2(b): G is of 2nd type.

Since G does not have any antipodal triplet, by Lemma 2.4.3, $\mathcal{A}(G,C)$ is isomorphic to an induced odd cycle $\alpha = G_1, G_2, \ldots, G_{2k+1}$, s.t. α is of 2nd type. Wlg, $W(G_2) \subset W(G_1)$. So by Lemma 2.4.3, G_1 is a strong separated graph and $W(G_1)$ is incomparable with $W(G_{i+1})$, $2 \le i \le 2k-1$. Let C_i be a principal clique of G_i, $1 \le i \le 2k+1$. Now $G_1 \Leftrightarrow G_2$, and $W(G_2) \subset W(G_1)$. So there exists C_1' in G_1 s.t. $C_2 > C_1'$. Note that $W(G_{2k+1})$ is not a subset of $W(G_1)$, otherwise C_1 will be a separating clique of G s.t. $\mathcal{A}(G,C_1)$ has more vertices than $\mathcal{A}(G,C)$



F_1 F_2

Figure 2.4.3: Some Forbidden Subgraphs For UV Graphs.

contradicting the choice of C. So $W(G_1)$ and $W(G_{2k+1})$ are incomparable. Assume that G_{2k+1} dominates some G_i, $i \notin \{1, 2k\}$. Then G_{2k+1} is a strong separated graph. Let $G'=G-(C-W(G_1))$. Then C_1 is a separating clique of G' and $\mathcal{A}(G',C_1)$ has an induced odd cycle, namely $G'_1,G'_2,\dots,G'_{2k+1}$, where G'_i is the separated graph of G' w.r.t. C_1 containing C_i, $2\le i\le 2k+1$, and G'_1 is the separated graph containing C'_1. Since $G'_{2k+1} \ge G'_i$, $2\le i\le 2k-1$, G' is not a path graph, which contradicts the fact that $G \in \mathcal{F}_{\mathcal{C}_2}$. So G_{2k+1} does not dominate any G_i, $1\le i\le 2k$, whence $W(G_{2k})$ is not a subset of $W(G_{2k+1})$. Hence $C_i \leftrightarrow C_{i+1}$, $2\le i\le 2k$. Let $x_i\in(C_i\cap C_{i+1})$, $y_i\in C_i-C$, $2\le i\le 2k$, $x_1\in (C_1\cap C'_1\cap C)$, $x_{2k+1}\in (C_{2k+1}\cap C)-C_1$, $y_1 \in(C'_1\cap C_1)-C$, and $y'_1 \in C'_1-C_1$. Then $G[\{x_1,x_2,\dots,x_{2k+1},y_1,y_2,\dots,y_{2k+1},y'_1\}]$ is isomorphic to to the graph $A_{11}(k>1)$ of last section.

Subcase 2(c): G is of 3rd type and G is not a bad forbidden subgraph.

By our assumption G does not contain any antipodal triplet. First assume that $\mathcal{A}(G,C)$ has an induced odd cycle of length at least 5. So condition (1) or (2) of Lemma 2.4.4 holds. Suppose $\mathcal{A}(G,C)$ is isomorphic to an induced odd cycle $\alpha=G_1,G_2,\dots,G_{2k+1}$, $k>1$. Now α contains at least one strong separated graph. So as in Theorem 2.3.6 subcase 3(b), G will be isomorphic to $A_{13}(k>1)$ of last section if α contains exactly one strong separated graph. In case α contains two strong separated graphs, then G will be isomorphic to $A_{14}(k>1)$.

So assume that there exist an odd cycle $\alpha=G_1,G_2,\dots,G_{2k+1}$ of $\mathcal{A}(G,C)$ and a separated graph G' s.t. $W(G_i) \subseteq W(G')$, $1\le i\le 2k$. Let $x\in C-C'$, where C' is a principal clique of G'. Let $x_i\in C_i\cap C_{i+1}$, and $y_i \in C_i-C$, where C_i is a principal clique of G_i, $1\le i\le 2k+1$. Let $y \in C'-C$. Let $F=G[\{x,y,x_1,x_2,\dots,x_{2l+1}, y_1,y_2,\dots,y_{2k+1}\}]$. Then F is isomorphic to $F_3(k)$ if G' predominates exactly 2k separated graphs. If G' predominates all the separated graphs of α, the F is isomorphic to $F_4(k)$. Since $F_3(k)$ and $F_4(k)$

are critical path graphs, G is isomorphic to $F_3(k>1)$ or $F_4(k>1)$, in this case.

Next assume that $\alpha = G_1, G_2, G_3$ is an odd cycle of $\mathcal{A}(G,C)$. Since G is not a bad forbidden subgraph, there exists a separated graph, say G_4 s.t. G_4 predominates at least two separated graphs of α. If G_4 predominates exactly two separated graphs of α, then G will be isomorphic to $F_3(1)$, otherwise G will be isomorphic to $F_4(1)$.∎

Conjecture: There is no bad minimal forbidden subgraphs for UV graphs.

The graph A_1 in Figure 2.3.3 shows that a strongly chordal graph is not necessarily a path graph. So it is natural to ask which strongly chordal graphs are path graphs. Since a strongly chordal graph does not have a trampoline as an induced subgraph, we have the following corollary to Theorem 2.4.5.

Corollary 2.4.6: A strongly chordal graph G is a path graph iff it contains neither F_1 of Figure 2.4.3 nor it contains any of the graphs A_1 to A_{15} other than A_7 and $A_{12}(k>1)$ as an induced subgraph.

Also every k-tree, k≥2 need not be a path graph. For example, the graph A_1 in Figure 2.3.3 is a 2-tree but not a path graph. In view of this it is natural to ask which k-trees, k≥2 are path graphs.

Let $\mathcal{C}_3^k$={G s.t. G is a k-tree and G is a path graph }, k≥2. Let $\mathcal{F}_{\mathcal{C}_3^k}$= {G $\notin \mathcal{C}_3^k$, but G−v $\in \mathcal{C}_3^k$ for every v $\in$ S(G)}. Note that $\mathcal{F}_{\mathcal{C}_3^k}$= { G $\notin$ $\mathcal{C}_3^k$, but G′ $\in \mathcal{C}_3^k$ for every sub k-tree G′ of G}. Next we characterize $\mathcal{F}_{\mathcal{C}_3^k}$.

Lemma 2.4.7: Let G $\in \mathcal{F}_{\mathcal{C}_3^k}$, k≥2. Then G has an antipodal triplet w.r.t. every separating clique C of G.

Proof: Let G_1, $G_2, \ldots, G_n$ be the separated subgraphs w.r.t. some separating clique C of G. Note that each G_i has exactly one principal clique. If there exist i and j s.t. $W(G_i) = w(G_j)$ then we claim that each of G_i and G_j has at least two relevant cliques. If possible, wlg, $W(G_1)=W(G_2)$ and G_1 has exactly one relevant clique. So $G_1 > G_2$. Let $G'=G-(C_1-C)$, where C_1 is the relevant clique of G. Now C is again a separating clique of G′ and G′is a sub k-tree of G. So G′ is a path graph and $\mathcal{A}(G',C)$ has a valid coloring. Now extend the coloring of $\mathcal{A}(G',C)$ to $\mathcal{A}(G,C)$ by assigning G_1 the color of G_2. The coloring so obtained is a valid coloring of $\mathcal{A}(G,C)$ because $G_i \leftrightarrow G_1$ implies $G_i \leftrightarrow G_2$. So a contradiction arises, and hence our claim holds. So $G_i \leftrightarrow G_j$ iff $i \neq j$ and $1 \le i,j \le r$. So every induced odd cycle of $\mathcal{A}(G,C)$ is of length 3. Let $\alpha=G_1$, G_2, G_3 be an induced odd cycle. If α is not of 3rd type then $\bigcap_{i=1}^{3} W(G_i) \neq \phi$ as $W(G_i) = |C|-1$, for all i, $1 \le i \le r$. So G has an antipodal triplet (x,G_1,G_2,G_3) where $x \in W(G_i)$ for all $1 \le i \le 3$. If r=3, then again G has an antipodal triplet as G is not a path graph. If possible, let G_4 be any other separated graph. Since $W(G_i)= |C|-1$ for all i, $1 \le i \le r$, $W(G_4)=W(G_i)$ for some i, $1 \le i \le 3$. WLG, $W(G_4) = W(G_1)$. Then (x,G_1,G_2,G_4) where $x \in W(G_1) \cap W(G_2) \cap W(G_4)$, is an antipodal triplet. So the sub k-tree $G'=G-(G_3-C)$ of G is not a path graph, a contradiction. So r=3 and hence G has an antipodal triplet. Also note that there exist i and j s.t. $W(G_i) \neq W(G_j)$, $1 \le i < j \le 3$; otherwise $\mathcal{A}(G,C_1)$ will have at least four separated graphs, which is a contradiction.∎

We next define some k-trees $M_1(k>2), M_2(k>2), \ldots, M_4(k>2)$ which will appear in the list of forbidden subgraphs for $\mathscr{C}_3^k$.

$M_1(k>2)$ is the k-tree with (k+4) vertices having three simplicial vertices, say x_1, x_2, and x_3 s.t. $N(x_i) \neq N(x_j)$ for $1 \le i < j \le 3$.

Let M be the k-tree on (k+4) vertices $\{v_1, v_2, \ldots, v_{k+1}, x_1, x_2, x_3\}$ s.t. $\{v_1, v_2, \ldots, v_{k+1}\}$ is a (k+1) clique and $N(x_1)=N(x_2)=\{v_2, v_3, \ldots, v_{k+1}\}$ and

$N(x_3)=\{v_1,v_2,v_4,\ldots,v_{k+1}\}$.

$M_2(k>2)$ is obtained from M by taking two new vertices y_1 and y_2 and making y_1 adjacent to all of $\{x_1,v_2,v_3,\ldots,v_k\}$, and y_2 adjacent to all of $\{x_2,v_2,v_3,\ldots,v_k\}$.

$M_3(k>2)$ is obtained from M by taking two new vertices y_1 and y_2 and making y_1 adjacent to all of $\{x_1,v_2,v_4,\ldots,v_{k+1}\}$, and y_2 adjacent to all of $\{x_2,v_2,v_4,\ldots,v_{k+1}\}$.

$M_4(k>2)$ is obtained from M by taking two new vertices y_1 and y_2 and making y_1 adjacent to all of $\{x_1,v_2,v_3,\ldots,v_k\}$, and y_2 adjacent to all of $\{x_2,v_2,v_4,\ldots,v_{k+1}\}$.

The graphs $M_1(3),\ldots,M_4(3)$ are illustrated in Figure 2.4.4.

Theorem 2.4.8: Let $G \in \mathcal{F}_{\mathcal{C}_3^k}$. Then either G is isomorphic to one of the graphs $M_1(k>2)$ to $M_4(k>2)$, or one of A_1 and A_2 of Figure 2.3.3.

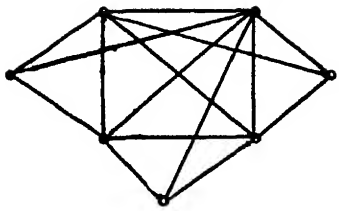
Proof: It is a routine exercise to check that each of the graphs mentioned in the Theorem belongs to $\mathcal{F}_{\mathcal{C}_3^k}$.

Necessity:

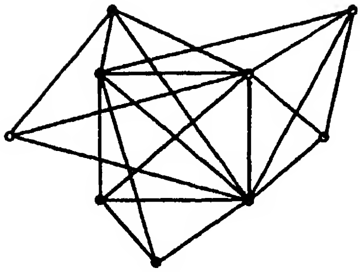
Let $G \in \mathcal{F}_{\mathcal{C}_3^k}$. Let C be a separating clique of G. By Lemma 2.4.7, G has an antipodal triplet w.r.t. C. Let G_1, G_2 and G_3 be the separated graphs.

CASE 1: $k>2$.

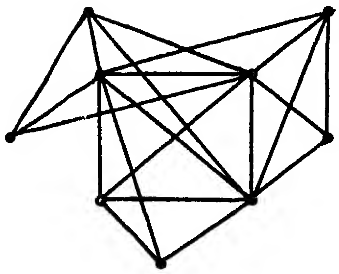
If $W(G_i) \neq W(G_j)$ for $i \neq j$, then G must be isomorphic to M_1. Otherwise wlg, let $W(G_1)=W(G_2)$. Clearly each of G_1 and G_2 has exactly one nonprincipal clique. Let C_1' and C_2' be the nonprincipal cliques of G_1 and G_2, respectively. Let $x \in C-W(G_3)$. Let $C_1' \cap C = C_2' \cap C$. If $x \in C_1' \cap C$, then G must be isomorphic to M_2 otherwise G will be isomorphic to M_3. Next assume that $C_1' \cap C \neq C_2' \cap C$. So either $x \in C_1' \cap C$ or $x \in C_2' \cap C$. So G will be isomorphic to M_4.



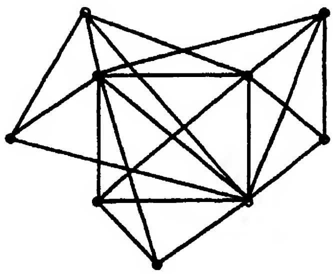
$M_1(3)$



$M_2(3)$



$M_3(3)$



$M_4(3)$

Figure 2.4.4: Illustration of Some k-trees which are not path Graphs

Case 2: $k=2$.

Wlg, let $W(G_1) = W(G_2)$. If $C_1' \cap C = C_2' \cap C = C_3 \cap C$, where C_1' and C_2' are some nonprincipal cliques of G_1 and G_2, respectively, and C_3 is a principal clique of G_3, then G must be isomorphic to A_1 of Figure 2.3.3, otherwise G will be isomorphic to A_2 of Figure 2.3.3.∎

A graph $G(V,E)$ is a split graph if $V=K \cup S$ s.t. $K \cap S=\phi$, K is a clique of G, and S is an independent set of G.

Let $\mathscr{C}_4$={ G s.t. G is a split graph as well as a path graph}. Since the graph A_8 in Figure 2.3.3 is a split and a critical path graph, we next find $\mathscr{F}_{\mathscr{C}_4}$, where $\mathscr{F}_{\mathscr{C}_4}$ = { H s.t. H is a split graph and H is a critical UV graph}.

Theorem 2.4.9: $G \in \mathscr{F}_{\mathscr{C}_4}$ iff G is isomorphic to one of the graphs $F_2, F_3(k)$, and $F_4(k)$ or one of the graphs $A_8, A_9, A_{13}(k>1)$, and $A_{14}(k>1)$.

Proof: It is easy to see that each of the graphs mentioned in the Theorem 2.4.9 belongs $\mathscr{F}_{\mathscr{C}_4}$. So we prove the necessity only. Since G is a split graph, $V(G)$ can be partitioned into two sets S_1 and S_2 s.t. $G[S_1]$ is an independent set and $G[S_2]$ is a clique.(see Golumbic[57]). Let C be a separating clique of G containing S_2. Then $C= S_2$ or $C= S_2 \cup \{x\}$, for some $x \in S_1$. Let $G_1, G_2, \ldots, G_r$ be the separated graphs of G w.r.t. C. Note that $|V(G_i)-C|=1$ for each $i=1, 2, \ldots r$. So $G_i \Leftrightarrow G_j$ iff $C_i \Leftrightarrow C_j$, where C_i and C_j are the only relevant cliques of G_i and G_j, respectively.

Case 1: G has an antipodal triplet w.r.t. C.

Let (x, G_1, G_2, G_3) be an antipodal triplet of G w.r.t. C. Now $\alpha=G_1, G_2, G_3$ is of 3rd type. So by Theorem 2.4.5, G will be isomorphic to one A_1 to A_{10} excluding A_7, or to F_2 of Figure 2.4.3. But A_8, A_9, and F_2 are the only split graphs.

Case 2: G has no antipodal triplet w.r.t. C.

Subcase 2(a): $\mathcal{A}(G,C)$ contains an induced odd cycle of length at least five.

Let $\alpha=G_1,G_2,\ldots,G_{2k+1}$, $k>1$, be an induced odd cycle of $\mathcal{A}(G,C)$. If α contains at least one strong separated graph, then as in Theorem 2.3.5, G will be isomorphic to $A_{14}(k>1)$ or $A_{13}(k>1)$ if α has exactly one strong separated graph or two strong separated graphs, respectively.

Subcase 2(b): No induced odd cycle of $\mathcal{A}(G,C)$ contains a strong separated graph.

Since G is not a path graph, there exists a separated graph say G_m s.t. G_m dominates some other separated graph. Let $\alpha=G_1,G_2,\ldots,G_{2k+1}$ be an induced odd cycle of $\mathcal{A}(G,C)$. If G_m is different from G_i, $1\le i\le 2k+1$, and G_m dominates at least 2k separated graphs of α, then as in Theorem 2.4.5, G will be isomorphic to one of $F_3(k)$ and $F_4(k)$. If possible $G_m=G_i$ for some i, $1\le i\le 2k+1$, say $G_m=G_1$. Assume that $k>1$. Now α has no strong separated graph. So wlg assume that $G_1|G_4$. Now C_1 is a separating clique of G but $C_4|C_1$, where C_1 and C_4 are the relevant cliques of G_1 and G_4, respectively. This is a contradiction to the Observation 2.4.1(ii). So clearly k=1. Let $x \in C-W(G_m)$. Now clearly C_m is a separating clique of G, where C_m is the principal clique of G_m. Let $G'_1,G'_2,\ldots,G'_r$ be the separated graphs of $\mathcal{A}(G,C_m)$ s.t. $x \in V(G'_r)$. If $\mathcal{A}(G,C_m)-G'_r$ does not contain any odd cycle then every odd cycle of $\mathcal{A}(G,C_m)$ is of 1st or 2nd type. So as in Theorem 2.4.5, G will be isomorphic to $A_{11}(k>1)$. But $A_{11}(k>1)$ is not a split graph. So $\mathcal{A}(G,C_m)-G'_r$ contains an odd cycle. So G will be isomorphic to $F_4(k)$ as $W(G'_i) \le W(G'_r)$ for all i, $1\le i\le r$. ▌

2.5 The Separator Theorem for Interval Graphs:

In this section we characterize interval graphs following the framework of Monma and Wei[92], and use this to find forbidden subgraphs for interval graphs in the subsequent section.

The following characterization of interval graphs as the intersection graphs of subpaths in a path, follows from the Theorem 1.4.5.

Theorem 2.5.1: A graph G is an interval graph iff there exists a path T s.t. $V(T)=\mathcal{C}(G)$ and for every vertex v of G, $T[\mathcal{C}_v(G)]$ is a subpath of T.

The path T satisfying Theorem 2.5.1 is called an interval clique tree of the interval graph G.

Let C separate G into separated graphs $G_1, G_2, \ldots, G_r$.

Proposition 2.5.2: If G is an interval graph, then each G_i is an interval graph with a clique tree T_i having C as an end vertex.

Proof: Let T be an interval clique tree for G. Let $\pi(V_i)$ be the subgraph of T consisting of vertices traversed by paths corresponding to the vertices in V_i, where $V_i=V(G_i)-C$. Since $G[V_i]$ is connected, so is $\pi(V_i)$. Since T is a path, $\pi(V_i)$ is a path. So there is a unique path $\pi^* = C, C_{i_1}, C_{i_2}, \ldots, C_{i_r}$ s.t. $C_{i_1}, \ldots, C_{i_{r-1}} \notin \pi(V_i)$ and $C_{i_r} \in \pi(V_i)$. Construct T_i by augmenting $\pi(V_i)$ by a new vertex C and a new edge $C\,C_{i_r}$. Then T_i is an interval clique tree for G_i having C as an end vertex.∎

A separated graph G_i is said to be nonrelevant if it has no nonrelevant cliques, otherwise it is called a relevant separated graph. Let $\mathcal{C}_i$ be the set of cliques of G_i excluding C. Let $\pi(\mathcal{C}_i) = T[\mathcal{C}_i]$. Then as we have seen in Proposition 2.5.2, $\pi(\mathcal{C}_i)=\pi(V_i)=T_i-C$. Let $\pi(v)$ denote the path (consisting of vertices) in T corresponding to $v \in V$. Since C is a separating clique of G, C corresponds to an internal vertex of T. Consider T as a rooted tree with C as the root. Then there are two branches, say B_1 and B_2, of T emanating from C.

We now characterize interval graphs in terms of separated graphs following the framework of Monma and Wei[92].

Theorem 2.5.3: G is an interval graph iff (1) each G_i is an interval graph,

and (2) the set S of separated graphs can be 2-colored s.t. no two antipodal graphs receive the same color, and that in each color class no two relevant cliques are unattached, and every separated graph , except possibly one, has no nonrelevant clique. The exceptional graph, should it exist, must be dominated by every separated graph of like color.

Proof: Necessity:

The condition (1) follows from Proposition 2.5.2. Let T be a clique tree of G. Color the separated graphs according to which branch of the tree it is on. Since T has two branches, G_i's are 2-colored. Assuming subgraphs $G_1 \leftrightarrow G_2$, and that G_1 and G_2 have the same color, we will find a contradiction.

There are two cases to consider.

Case 1: $C_1 \leftrightarrow C_2$ for some C_i in G_i, i=1,2.

Let $x \in (C_1 \cap C) \setminus C_2$ and $y \in (C_2 \cap C) \setminus C_1$. Now $\pi(x)$ and $\pi(y)$ are paths of T each containing the vertex C of T. Since G_1 and G_2 have the same color, either $\pi(x)$ is a subpath of $\pi(y)$ or conversely. Wlg, $\pi(x)$ is a sub path of $\pi(y)$. Since $y \notin C_1$ and $\pi(y)$ contains C_1, T is not an interval clique tree for G, a contradiction.

Case 2: $C_1 > C_2$, $C_1' < C_2'$ for some C_1, C_1' in G_1 and C_2, C_2' in G_2.

Let $x \in (C_1 \cap C) \setminus C_2$ and $y \in (C_2' \cap C) \setminus C_1'$. Wlg, $\pi(x)$ is a subpath of $\pi(y)$. So $\pi(C_1)$ is a subpath of $\pi(y)$. This yields a contradiction as $y \notin C_1'$. So no two antipodal separated graphs receive the same color.

We claim that no two relevant cliques of the separated graphs with the same color are unattached. If possible, let $C_1 | C_2$, where $C_i \in G_i$, i=1,2, and G_1 and G_2 have the same color(G_1 and G_2 may be the same separated graph). Let $x \in (C_1 \cap C)$ and $y \in (C_2 \cap C)$. Then $\pi(x)$ is a subpath of $\pi(y)$ or conversely. Since $x \notin C_2$ and $y \notin C_1$, T is an interval clique tree for G is contradicted.

Now merge the vertex C of $T^{(1)}$ and the vertex C of $T^{(2)}$ to obtain a tree T. It is easy to see that T is an interval clique tree for G. Hence G is an interval graph.∎

2.6 Forbidden Subgraph Characterization of Interval Graphs:

In this section we present the forbidden subgraph characterization for interval graphs. Let G be a chordal graph and let $G_1, G_2, \ldots, G_r$ be the separated graphs of G w.r.t. some separating clique C. G_i is said to be compatible with G_j if either $G_i > G_j$ and (i) G_i is a nonrelevant separated subgraph, and (ii) no relevant clique of G_i is unattached with any relevant clique of G_j, or G_i is equivalent to G_j and at least one is a nonrelevant separated graph. Otherwise G_i is said to be noncompatible with G_j. Define the graph $\mathcal{B}(G,C)$ by $V(\mathcal{B}(G,C))=\{G_i,\ 1\le i\le r\}$, and $E(\mathcal{B}(G,C)) = \{G_iG_j,\ 1 \le i,j \le r$, s.t. G_i and G_j are noncompatible$\}$.

The following theorem gives the structure of $\mathcal{B}(G,C)$.

Theorem 2.6.1: Let G be a critical interval Graph. Then the number of separated graphs w.r.t. any separating clique is 3.

Proof: Let C be any separating clique of G. Since G is a critical interval graph, $\mathcal{B}(G,C)$ is nonbipartite. So $\mathcal{B}(G,C)$ contains an induced odd cycle, say $\alpha = G_1, G_2, \ldots, G_{2k+1}$. Since G is critical interval graph $\mathcal{B}(G,C)$ must be isomorphic to α. We claim that k=1. Let $W(G_1)$ be a maximal set amongst $W(G_i)$'s. Since α is an induced odd cycle, $W(G_i) \subseteq W(G_1)$, $2 < i < 2k+1$. Again if G_1 is a relevant separated graph, then $W(G_i) = W(G_1)$, $2 < i < 2k+1$. As G_1 is compatible with G_i, $2<i<2k+1$, no G_i is a relevant separated graph except possibly i=1, 2 , 2k+1. So wlg, we may take G_1 to be nonrelevant. Now if $G_1 | G_2$ or $G_1 | G_{2k+1}$, then k=1 as $G_1|G_j$ implies $G_1|G_4$ for j=2 or 2k+1, and hence α will have a chord, a contradiction! Again G_2 is compatible with G_{2k+1}. So assume that $W(G_{2k+1}) \subseteq W(G_2)$.

Case 1: $W(G_{2k+1})$ is not a subset of $W(G_1)$.

Then clearly $G_1 \leftrightarrow G_2$ and $G_1 \leftrightarrow G_{2k+1}$. If possible let $W(G_2)=W(G_{2k+1})$. If G_2 is equivalent with G_{2k+1}, then G_2 is noncompatible with G_{2k} as G_{2k+1} is noncompatible with G_{2k}. So a contradiction arises. So assume wlg that $G_2 > G_{2k+1}$. Since $W(G_2)= W(G_{2k+1})$, G_2 has no nonprincipal clique. So G_{2k+1} is noncompatible with G_3, as G_2 is noncompatible with G_3, which is a contradiction. Thus $W(G_{2k+1}) \subset W(G_2)$. Let $x \in W(G_2)-W(G_1)$ and $y \in W(G_{2k+1})-W(G_1)$. Let $G'=G-y$. Let C' be a principal clique of G_1. Then C' is a separating clique of G'. Let $G_i(1)$ be the separated graph of G' w.r.t. C' containing $V(G_i)-W(G_i)(W(G_i)$ is w.r.t. $C)$, $2<i<2k+1$. Let G_1' be the separated graph of G' containing $V(G_2)-W(G_2)$. Then $\alpha'=G_1'$, $G_3(1)$, $G_4(1),\ldots,G_{2k}(1)$ is an odd cycle of $\mathcal{B}(G',C')$. So G' is not an interval graph, which is a contradiction.

Case 2: $W(G_{2k+1}) \subset W(G_1)$.

Since G_1 is noncompatible with G_{2k+1}, either there exists a clique C_1'' of G_1 s.t. $C_1'' \mid C_{2k+1}$ or $C_{2k+1} > C_1''$.

Subcase 2(a): $C_1'' \cap C_{2k+1} = \phi$.

Now G_3 is compatible with G_{2k+1}. So $W(G_3) \cap W(G_{2k+1}) \neq \phi$..(1) But G_1 is compatible with G_3 and G_1 dominates G_3, so $W(G_3) \subseteq C_1'' \cap C$. Since $C_1'' \cap C_{2k+1} = \phi$, $W(G_3) \cap W(G_{2k+1}) = \phi$, which is a contradiction to (1).

Subcase 2(b): $C_1'' \cap C_{2k+1} \neq \phi$.

Clearly $C_1'' \cap C \subset W(G_{2k+1})$. If $W(G_2) \subseteq W(G_1)$, then let $x \in C-W(G_1)$, otherwise let $x \in W(G_2)-W(G_1)$. Let $G'=G-x$. Let C_1' be a principal clique of G_1. Then C_1' is a separating clique of G'. If $W(G_2) \subseteq W(G_1)$, then $G_2(1)$, $G_3(1),\ldots$, $G_{2k+1}(1)$, G_1', where G_1' is the separated graph containing the clique C_1'', is an odd cycle of $\mathcal{B}(G',C_1')$. So G' is not an interval graph, a contradiction. If $W(G_2)$ is not a subset of $W(G_1)$, then let $\alpha'=G_2'(1)$, $G_3(1)$, $G_4(1),\ldots$, $G_{2k+1}(1)$, G_1', where $G_2'(1)$ is the separated graph of G' containing G_2-x, and G_1' is the separated graph containing the clique $C_1''-C$.

Then as in case 1, α' is an odd cycle of $\mathcal{B}(G',C_1')$. So G' is not an interval graph, a contradiction. So $k=1$.█

Lemma 2.6.2: Let G be critical interval graph. If G has a relevant separated graph w.r.t. some separating clique, then G has three separated graphs G_1',G_2', and G_3' s.t. $G_1'|G_j'$, $j=2,3$.

Proof: Let C be a separating clique of G s.t. G has a relevant separated graph G_1 with minimum number of cliques. Since G is a critical interval graph, G has three separated graphs w.r.t. to C. Let G_2 and G_3 be the other separated graphs. Since G_1 is a relevant separated graph, either $G_1 > G_2$ or $G_1 > G_3$. Wlg, $G_1 > G_2$. Let C_1 be a principal clique of G_1. Then C_1 is a separating clique of G. Let G_i', $1\le i\le 3$ be the separated graphs of G w.r.t. C_1 s.t. G_i' contains $G_1-W(G_1)$. $(W(G_1)$ is w.r.t. C). Then Clearly either $G_1'|G_3'$ or $G_1'\Leftrightarrow G_3'$. If $G_1'|G_3'$, then $G_2'|G_3'$, and our lemma is true. So assume that $G_1'\Leftrightarrow G_3'$. Then clearly G_1' is a relevant separated graph of G s.t. G_1' has less number of cliques than that of G_1. This contradicts the choice of C and proves the lemma.█

Theorem 2.6.3: G is an interval graph iff G does not contain any of the graphs in Figure 2.6.1 as induced subgraph.

Proof: It is a routine exercise to see that none of the graphs in Figure 2.6.1 is an interval graph. So the sufficiency.

Necessity:

If possible, G is not an interval graph. Wlg, let G be a critical interval graph. If G is not chordal then G must be isomorphic to I_1 of Figure 2.6.1. So let G be chordal.

Case 1: G has separated graphs G_1,G_2, and G_3 w.r.t. some separating clique C s.t. $G_1|G_i$, $i=2,3$.

Subcase 1(a): Exactly one of G_2 and G_3 is a relevant separated graph.

Wlg, G_2 is a relevant separated graph. Then clearly $G_2 \ge G_3$. If $W(G_3)$

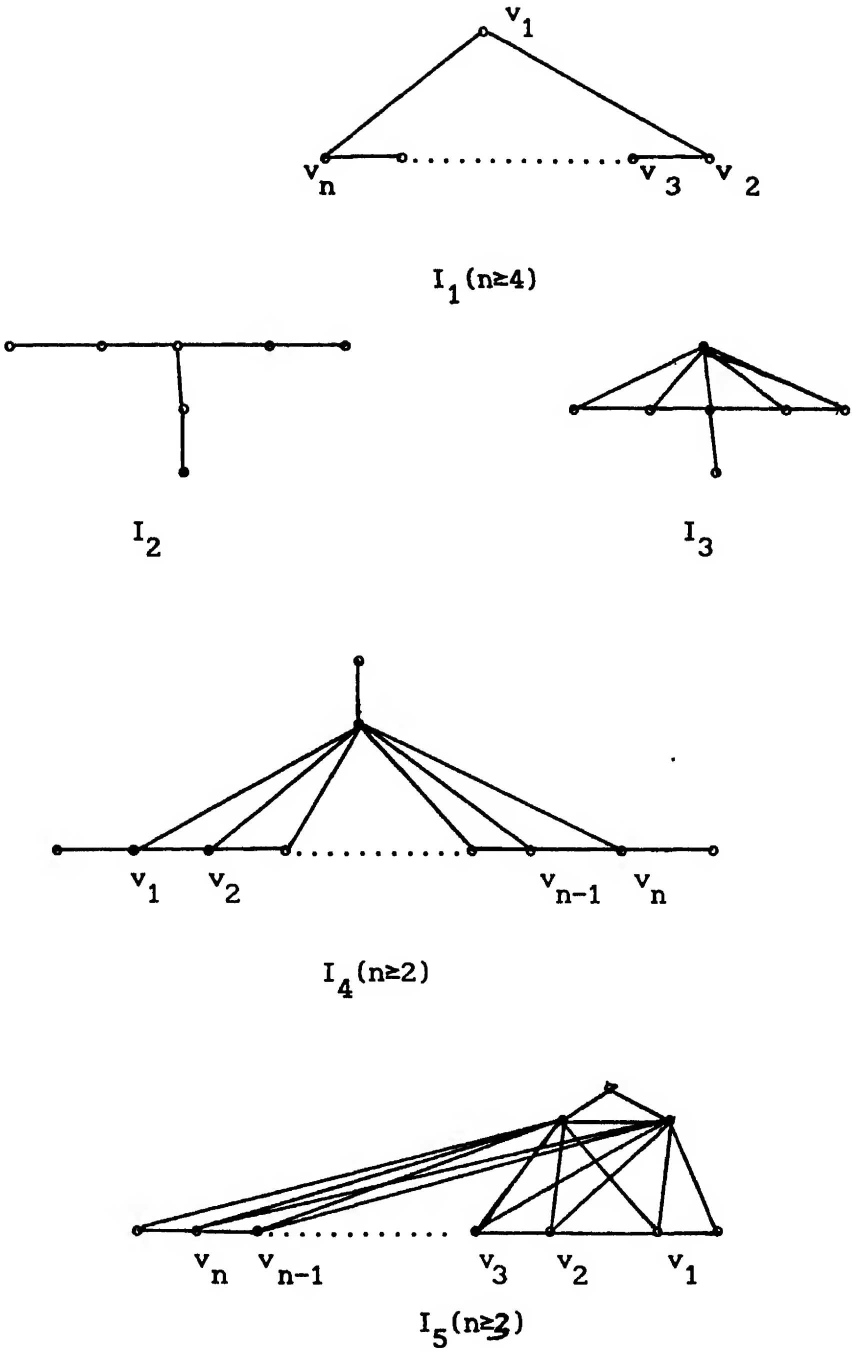


Figure 2.6.1: Forbidden Subgraphs For Interval Graphs.

$\subset W(G_2)$, then let $x \in W(G_2) \cap W(G_3)$, $y \in W(G_1)$, and $z \in W(G_2)-W(G_3)$. Let C_2' be a nonrelevant clique of G_2. Let y' be a vertex of C_2' which is not adjacent to any vertex of C. Let $P=y',y_1,y_2,\ldots,y_r,z$ be a y'-z shortest path in $G_2-(W(G_2)-\{z\})$. Let $z_1 \in C_3-C$ and $z_2 \in C_1-C$, where C_i is a principal clique of G_i, $i=1,3$. Then $G[\{x,y,z,z_1,z_2,y',y_1,\ldots,y_r\}]$ is isomorphic to $I_4(n\geq2)$. Since $I_4(n\geq2)$ is a critical interval graph, G is isomorphic to $I_4(n\geq2)$.

If $W(G_2)=W(G_3)$, then G_3 has a nonprincipal clique C_3'. Let $z \in C_3-C_3'$, $z_1 \in W(G_1)$, and $z_2 \in C_1-C$, where C_1 is a principal clique of G_1. Let $P_1=z,y_1,y_2$ be an induced path of length 2 of $G_2-(W(G_2)-z)$, and $P_2=z,y_1,y_2$ be an induced path of length 2 of $G_3-(W(G_3)-z)$. The existence of P_1 and P_2 is assured since G_2 has more than two cliques and G_3 has a nonprincipal clique. Then $G'=G[\{z,z_1,z_2,x_1,x_2,y_1,y_2\}]$ is isomorphic to I_3, a critical interval graph. Since G' is a proper induced subgraph of G, we have a contradiction. So $W(G_2) \neq W(G_3)$.

Subcase 1(b): Both G_2 and G_3 are relevant separated graphs of G.

So G_2 is attached to G_3. Let $x \in W(G_2) \cap W(G_3)$. Then there exists an induced path of length 2, say x,x_1,x_2 starting from x in $G_2-(C-\{x\})$, as G_2 has a nonrelevant clique. Similarly there exists an induced path of length 2, say x,y_1,y_2 in $G_3-(C-\{x\})$. Let $y \in W(G_1)$, and $y' \in C_1-C$, where C_1 is a principal clique of G_1. Then $G[\{x,y,x_1,x_2,y_1,y_2,y'\}]$ is isomorphic to I_3. Since I_3 is a critical interval graph, G is isomorphic to I_3 in this case.

Subcase 1(c): Neither G_2 nor G_3 is a relevant separated graph.

First assume that $G_2|G_3$. Let $x_i \in W(G_i)$, and $y_i \in C_i-C$, where C_i is a principal clique of G_i, $1\leq i\leq3$. Then $G[\{x_1,x_2,x_3,y_1,y_2,y_3\}]$ is isomorphic to $I_4(n=2)$. Since $I_4(n=2)$ is a critical interval graph, G is isomorphic to $I_4(n=2)$.

Let $G_2 \Leftrightarrow G_3$. So by Lemma 2.3.5, $G_2 \cup G_3$ will contain a subgraph isomorphic to one of H_1' to H_5'. Let $z_1 \in W(G_1)$, and $z_2 \in C_1-C$, where C_1 is a

principal clique of G_1. Then clearly $G[(G_2 \cup G_3 \cup \{z_1, z_2\})]$ is not a critical interval graph. So G_2 is not antipodal to G_3.

So $W(G_2)$ and $W(G_3)$ are comparable. Clearly G_2 is not congruent to G_3. Wlg, $G_2 > G_3$, and $W(G_3) \subset W(G_2)$. So there exists a relevant clique C_2' of G_2 and a relevant clique C_3' of G_3 s.t. $C_2'|C_3'$. Since $W(G_3) \subset W(G_2)$, and G_2 is not antipodal to G_3, $C_2'|C_3$. Let $x_1 \in C_2'-C$ s.t. x_1 is not adjacent to any vertex of $W(G_3)$. Let $G_2' = G_2-(V(G_2)-(C\cup(x_1))$, and $G'=G-(V(G_2)-(C\cup\{x\}))$. Then G_1, G_2', and G_3 are pair wise unattached separated graphs of G' w.r.t.C. So G' is not an interval graph. Hence a contradiction.

Case 2: There exist no separated graphs G_1, G_2, and G_3 s.t. $G_1|G_i$, i=2,3.

Let G_1, G_2, G_3 be the separated graphs w.r.t. some separating clique C. Then by Lemma 2.6.2, no G_i is a relevant separated graph, $1 \le i \le 3$.

Subcase 2(a): G_1, G_2, and G_3 are pair wise antipodal.

Then G is a critical DV graph. So by Theorem 2.3.6 G is isomorphic to one of A_1 to A_{10} in Figure 2.3.3. Since A_6 and A_7 are the only critical interval graphs among A_1 to A_{10}, G is isomorphic to $I_5(n \ge 2)$.

Subcase 2(b): $G_1 \Leftrightarrow G_2$, $G_2 \Leftrightarrow G_3$ and $G_1|G_3$. Let C_i be a principal clique of G_i, $1 \le i \le 3$. Since $G_1|G_3$, neither $W(G_2) \le W(G_1)$ nor $W(G_2) \le W(G_3)$. We claim that neither $W(G_1) \le W(G_2)$ nor $W(G_3) \le W(G_2)$. First note that either $W(G_1)$ is not a subset of $W(G_2)$ or $W(G_3)$ is not a subset of $W(G_2)$, otherwise $\mathcal{A}(G,C_2)$ will have at least four separated graphs, contradicting Theorem 2.6.1. Now wlg, assume that $W(G_1) \le W(G_2)$. Let $C'=W(G_2) \cap W(G_3)$, and $G'=G-C'$. Then $G_i'=G_i-C'$, $1 \le i \le 3$ are the separated graphs of G' w.r.t. $C-C'$ s.t. $G_1' \Leftrightarrow G_2'$ and $G_1'|\, G_i'$, i=2,3. So by Theorem 2.5.3, G' is not an interval graph. This contradicts the fact that G is a critical interval graph. So our claim is true. So $W(G_i)$'s are pair wise incomparable. Hence $C_1 \Leftrightarrow C_2$, $C_2 \Leftrightarrow C_3$, and $C_1|C_3$. Let $x_1 \in W(G_1)-W(G_2)$, $x_2 \in W(G_2) \cap W(G_1)$, $x_3 \in W(G_2) \cap W(G_3)$, $x_4 \in W(G_3)-W(G_2)$, and $y_i \in C_i-C$, $1 \le i \le 3$. Then

$G[\{x_1,x_2,x_3,x_4,y_1,y_2,y_3\}]$ is isomorphic to $I_5(n=2)$. Since $I_5(n=2)$ is a critical interval graph, G is isomorphic to $I_5(n=2)$ in this case.

Subcase 2(c): There is exactly one pair of antipodal subgraphs.

Wlg, $G_1 \leftrightarrow G_2$, and G_3 is attached to either G_1 or to G_2. Let C_i be a principal clique of G_i, $1 \le i \le 3$. If $G_3 \ge G_1$ and $G_3 \ge G_2$, then $\mathcal{B}(G,C_3)$ will have at least four vertices as G_3 will then have at least two relevant cliques, which contradicts Theorem 2.6.2. So $W(G_3) \subseteq W(G_1) \cup W(G_2)$.

Assume that $W(G_1)$ and $W(G_2)$ are comparable. If $W(G_1) \subseteq W(G_2)$, then take $C'=C_2$, otherwise take $C'=C_1$. Then $\mathcal{A}(G,C')$ will have at least four vertices, which again contradicts Theorem 2.6.2. So our assumption is wrong. Hence $C_1 \leftrightarrow C_2$. If $G_3 \ge G_1$, then as G_3 does not dominate G_2, $G_3 \leftrightarrow G_2$, which is a contradiction.

Now wlg, assume that $G_1 \ge G_3$. We claim that $W(G_2) \cap W(G_3) = \phi$. Assume that our claim is not true. So $G_2 \ge G_3$ as G_3 does not dominate G_2, since $G_1 \ge G_3$ and $G_1 \leftrightarrow G_2$. So there exists C_i' in G_i, s.t. $C_3 \mid C_i'$, $i=1,2$. Then clearly C_2 is a separating clique of G. Let G_1', G_2', and G_3' be the separated graphs of G w.r.t. C_2 s.t. G_i' contains $V(G_i)-W(G_i)$, $1 \le i \le 3$. Then clearly $G_1' \leftrightarrow G_2'$, $G_1' \ge G_3'$, and $G_2' | G_3'$. So wlg we can assume that our claim is true.

So $G_3 | G_2$. Since no separated graph is relevant, there exists a clique C_1' of G_1 s.t. $C_1' | C_3$. we claim that C_1' is attached to C_2. Assume $C_1' | C_2$. Let $C'=C-(W(G_1) \cap W(G_2))$, and $G'=G-(W(G_1) \cap W(G_2))$. Then $G_i'=G_i-C'$ is a separated subgraph of G', $1 \le i \le 3$, s.t. $G_1' | G_2'$, $G_3' | G_2'$, and $C_1' | W(G_3')$. So G' is not an interval graph, whence C_1' is attached to C_2.

Let $x \in C_1' \cap C_2$, $y \in W(G_1) \cap W(G_3)$, and $z \in W(G_2)-W(G_1)$. Let $x_1 \in C_1'-C_1$, and $y_1 \in C_2-C$. Let P_1 be an x_1-y shortest path in $G_1-(C-y)$. Then clearly P_1 is of length at least 2. Let $y_1 \in C_2-C$. Let x_1',x_2',y be a section of P_1. Let $z_1 \in C_3-C$. Then $G[\{x,y,z,z_1,x_1',x_2',y_1\}]$ is isomorphic to I_2. Since I_2 is a critical interval graph, G is isomorphic to I_2 in this case.

Subcase 2(d): There is no pair of antipodal subgraphs.

Let $W(G_1)$ be a maximal set. Since there exists no G_i s.t. $G_i|G_j$, and $G_i|G_k$, $1 \le i,j,k \le 3$, $G_1 \ge G_2$ and $G_1 \ge G_3$. Also G_1 has at least two relevant cliques C_1 and C_1', where C_1 is a principal clique of G_1. Then clearly G has at least four separated graphs w.r.t. the separating clique C_1, which contradicts Theorem 2.6.1.

So G must be isomorphic to one of the graphs I_1 to I_5. Since we have taken a critical interval graph, our Theorem is proved.∎

2.7 Forbidden Subgraphs For Proper Interval Graphs:

In this section, we characterize proper interval graphs in terms of separated graphs and then find out all the minimal forbidden subgraphs for proper interval graphs.

A family F of intervals is said to be a proper interval representation, or P.I.R., of a proper interval graph G if G is the intersection graph of F, and no interval in F properly contains another interval in F. For any interval $I=[a,b]$, let $L(I)=a$ and $R(I)=b$. For a collection of intervals F, let $\min(L(F))= \min\{L(I) \text{ s.t. } I \in F\}$, and $\max(R(F))=\max\{R(I) \text{ s.t. } I \in F\}$.

Before presenting characterization of proper interval graphs, we first prove the following lemma.

Lemma 2.7.1: Let $C=\{v_1,v_2,\ldots,v_s\}$ be a nonseparating clique of a proper interval graph G s.t. $S(G) \cap C \ne \phi$. Let $X_1=\{v_1,v_2,\ldots,v_r\}$ be the subset of C s.t. v_i lies in exactly one clique of G, $1 \le i \le r$. Let $X_2=\{v_{r+1},v_{r+2},\ldots,v_t\} \subset C$ be s.t. v_j lies in exactly two cliques of G, $r+1 \le j \le t$, and $X_3=\{v_{t+1},\ldots,v_s\}$. Then in any P.I.R. of G the intervals corresponding to the vertices of C occur consecutively. Furthermore, (i) there exists a P.I.R. of G satisfying the following: $L(I_{v_1}) < L(I_{v_2}) < \ldots < L(I_{v_s}) < L(I_y)$ for every $y \in V(G)-C$,

(ii) There exists a P.I.R. satisfying the following: $R(I_y) < R(I_{v_s}) < R(I_{v_{s-1}}) < \dots < R(I_{v_1})$, for every $y \in V(G)-C$.

Proof: Let $F=\{I_v,\ v \in V(G)\}$ be P.I.R. of a proper interval graph G. let $p \in \bigcap_{i=1}^{s} I_{v_i}$. If possible, let $y \in V(G)$ s.t. $L(I_{v_i}) < L(I_y) < L(I_{v_j})$ for some i,j, $1 \le i,j \le s$. Then $L(I_y) < L(I_{v_j}) < p$, and $R(I_y) > R(I_{v_i}) > p$. So $C \cup \{y\}$ is a clique of G, which contradicts the maximality of C. So the intervals corresponding to the vertices of C occur consecutively. Since C is not a separating clique of G, for every P.I.R. F of G, either $L(F)=L(\{I_v,\ v \in C\})$ or $R(F)=R(\{I_v, v \in C\})$. Let $L(F)=L(\{I_v, v \in C\})$. Since each vertex of X_1 lies in exactly one clique of G, we can construct a P.I.R. F_1 of G from F by interchanging the intervals if needed, satisfying the following: $L(I_{v_1}) < L(I_{v_2}) \dots < L(I_{v_r})$. Since each vertex of X_2 lies in exactly two cliques of G, and the intervals corresponding to the vertices of C occur consecutively, we can construct a P.I.R. F_2 of G from F_1 satisfying (i).

We construct a P.I.R. F_3 from F_2 as follows:

Let $a=\mathrm{Min}(L(F_2))$. Wlg, we take $a=0$. Let $F_3=\{\ I'=[-d,-c],\ I=[c,d] \in F_2\}$. Clearly F_3 is a P.I.R. of G satisfying Lemma 2.7.1 (ii).

If $R(F)=R(\{I_v, v \in C\})$, then wlg, assume that $L(F)=0$. Now $F'=\{I'=[-d,-c]$, where $I=[c,d] \in F\}$. Then F' is a P.I.R.of G s.t. $L(F')=L(\{I_v,\ v \in C\})$. Now as above we can construct the required P.I.R.s.∎

Theorem 2.7.2: (Separator Theorem) Let $G_1, G_2, \dots, G_r$, $r \ge 2$ be the separated graphs w.r.t. a separating clique C. Then G is a proper interval graph iff

(i) each G_i is a proper interval graph,

(ii) If $W(G_1) \cap W(G_2) \ne \phi$, then $W(G_1) \cup W(G_2) = C$, and there is exactly one clique C_i in G_i intersecting $W(G_1) \cap W(G_2)$, $i=1,2$, and

(iii) $r=2$.

Proof: Necessity:

(i) This follows from the fact that every induced subgraph of a proper interval graph is a proper interval graph.

(ii) If possible let there be G_1 and G_2 s.t. $\phi \neq W(G_1) \cap W(G_2)$, and $W(G_1) \cup W(G_2) \neq C$. Let $x \in W(G_1) \cap W(G_2)$, $x_1 \in C \setminus (W(G_1) \cup W(G_2))$, $y_1 \in C_1 \setminus W(G_1)$, and $z_1 \in C_2 \setminus W(G_2)$, where C_i is a principal clique of G_i, i=1,2. Let I_x, $I_{x_1}, I_{y_1}, I_{z_1}$ be the intervals corresponding to x, x_1, y_1, and z_1, respectively in some P.I.R. F of G. Wlg, $R(I_{x_1}) < L(I_{y_1})$ and $R(I_{y_1}) < L(I_{z_1})$. Then as $I_x \cap I_{x_1} \neq \phi$, $I_x \cap I_{z_1} \neq \phi$, $I_{y_1} \subset I_x$ which contradicts the fact that F is a P.I.R. of G.

Suppose $\phi \neq W(G_1) \cap W(G_2)$ and $W(G_1) \cup W(G_2) = C$ but there are two relevant cliques C_1 and C_1' of G_1 intersecting $W(G_1) \cap W(G_2)$. Let $x_1 \in C_1 - C_1'$, and $y_1 \in C_1' - C_1$, $x \in W(G_1) \cap W(G_2)$, and $z_1 \in C_2 - W(G_2)$, where C_2 is a relevant clique of G_2. Now x is adjacent to all of x_1, y_1, and z_1, and $\{x_1, y_1, z_1\}$ is an independent set in G. So as above, G is not a proper interval graph.

(iii) Suppose $r \geq 3$. If $W(G_1)$, $W(G_2)$, and $W(G_3)$ are pair wise disjoint, then by Theorem 2.5.3, G is not an interval graph, and hence not a proper interval graph. If $W(G_i) \leq W(G_j)$ for $1 \leq i,j \leq 3$, then $W(G_i) \cap W(G_j) \neq C$ which contradicts Theorem 2.7.2 (ii). So $W(G_1)$, $W(G_2)$, and $W(G_3)$ are pairwise not incomparable, and $W(G_i) \cap W(G_j) \neq \phi$, $1 \leq i,j \leq 3$. So G_1, G_2, and G_3 are pairwise antipodal. So by Theorem 2.5.3, G is not an interval graph, and hence not a proper interval graph.

Sufficiency:

Assume that the separated graphs satisfy the conditions of Theorem 2.7.2. We claim that G is a proper interval graph.

Case 1: $G_1 | G_2$.

Let $W(G_1) = \{v_1, v_2, \dots, v_r\}$, $W(G_2) = \{v_t, v_{t+1}, \dots, v_s\}$, and $X = C - (W(G_1) \cup$

$W(G_2)) = \{ v_{r+1},\dots,v_{t-1}\}$. Now C is not a separating clique of G_i, i=1,2. Again $C \cap S(G_1)=\{v_{r+1},\dots,v_s\}$. So by Lemma 2.7.1, there is a P.I.R. F_1 of G_1 satisfying the following: $R(I_y) < R(I_{v_1}) < R(I_{v_2}) \dots < I(I_{v_s})$ for every $y \in V(G_1)-C$. Again by Lemma 2.7.1, there is a P.I.R. F_2 of G_2 s.t. $L(I_{v_1}) < L(I_{v_2}) < \dots < L(I_{v_s}) < L(I_y)$ for every $y \in V(G_2)-C$. Wlg, no intervals in F_1 intersect any interval in F_2. Let $l_i = L(I_{v_i})$ where $I_{v_i} \in F_1$, and $r_i=R(I_{v_i})$ where $I_{v_i} \in F_2$. Let $F_3= ((F_1 \cup F_2)-\{I_{v_i},\ v_i \in C\}) \cup (\{I'_{v_i}=[l_i,r_i],\ v_i \in C\})$. Then F_3 is a P.I.R. of G, and hence G is a proper interval graph.

Case 2: $W(G_1) \cap W(G_2) \neq \phi$.

Let $X= \{v_1,v_2,\dots,v_r\}= W(G_1)-W(G_2)$. $Y= \{v_{r+1},\dots,v_t\}= W(G_1) \cap W(G_2)$, and $Z =\{v_{t+1},v_{t+2},\dots,v_s\}$. Then by Lemma 2.7.1,there is a P.I.R. F_1 of G_1 satisfying the following: $R(I_y) < R(I_{v_1}) < R(I_{v_2}) \dots < I(I_{v_s})$ for every $y \in V(G_1)-C$. Again by Lemma 2.7.1, there is a P.I.R. F_2 of G_2 s.t. $L(I_{v_1}) < L(I_{v_2}) < \dots < L(I_{v_s}) < L(I_y)$ for every $y \in V(G_2)-C$. Wlg, no intervals in F_1 intersect any interval in F_2. Let $l_i = L(I_{v_i})$ where $I_{v_i} \in F_1$, and $r_i=R(I_{v_i})$ where $I_{v_i} \in F_2$. Let $F_3= ((F_1 \cup F_2)-\{I_{v_i},\ v_i \in C\}) \cup (\{I'_{v_i}=[l_i,r_i],\ v_i \in C\})$. Then F_3 is a P.I.R. of G, and hence G is a proper interval graph.▌

Theorem 2.7.3: (Forbidden subgraph characterization) A graph G is a proper interval graph iff G does nor contain any of the graphs in Figure 2.7.1 as an induced subgraph.

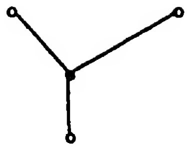
Proof:

Necessity:

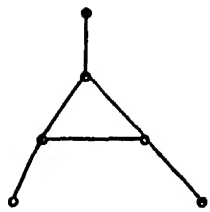
Clearly none of the graphs in Figure 2.7.1 is a proper interval graph.

Sufficiency:

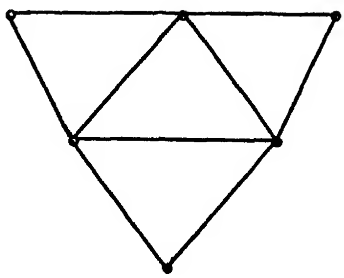
Assume G is not a proper interval graph and every induced subgraph of G is a proper interval graph. If G is not chordal then G is isomorphic to a C_n, $n \geq 4$, which is P_4, as every proper induced subgraph of G is a proper



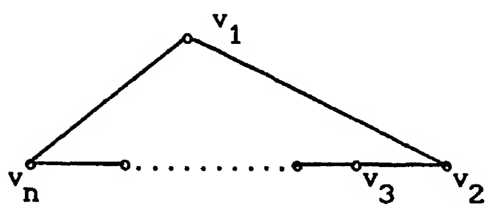
P_1



P_2



P_3



$P_4(n \geq 4)$

Figure 2.7.1: Forbidden Subgraphs For Proper Interval Graphs.

interval graph. Assume that G is chordal. Again it has a separating clique C. Let $G_1, \ldots, G_r$, $r \geq 2$, be the separated graphs of G w.r.t. C. So Theorem 2.7.2 will not be satisfied. If Theorem 2.7.2 (ii) is violated, then G will be isomorphic to P_1, and if Theorem 2.7.2 (iii) is not true then G will be isomorphic to either P_2 or P_3, as we have seen in the proof of the Theorem 2.7.2.∎

2.8 Forbidden Subgraphs for Chordal Planar Graphs:

In this section we characterize chordal planar graphs following the framework of Monma and Wei [92] and then present the forbidden subgraph characterization for this class.

Let P be a planar representation of planar graph G. Let ex(P) denote the exterior face of P, and $\delta(f)=\{q \in R^2$ s.t. $q \in$ boundary of the face $f\}$.

Lemma 2.8.1: Let G_1 and G_2 be two planar graphs. If either $|V(G_1)\cap V(G_2)| = 1$ or $V(G_1) \cap V(G_2) = \{v_1,v_2\}$ with $v_1v_2 \in E(G_1)\cap E(G_2)$, then $(G_1 \cup G_2)$ is planar.

Proof: Trivial.

Let $G_1, G_2, \ldots, G_r$ be the separated graphs of G w.r.t. C. Let R(G,C) = { G_i s.t. $|W(G_i)| \geq 3\}$.

Theorem 2.8.2: (Separator Theorem) A chordal graph G is planar iff (i) each G_i is planar, and

(ii) G_1, $G_2 \in R(G,C)$ implies $W(G_1) \neq W(G_2)$.

Proof: Necessity:

(i) This follows from the fact that every induced subgraph of a planar graph is planar.

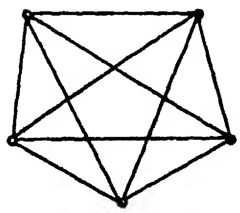
(ii) Let $\{v_1,v_2,v_3\} \subseteq W(G_1)=W(G_2)$ for $G_1,G_2 \in R(G,C)$. Let $x_i \in C_i-C$, where C_i is a principal clique of G_i, i=1,2 and $x_3 \in C-W(G_1)$, then $G[\{v_1,v_2,v_3,x_1,x_2,x_3\}]$ is nonplanar as it contains $K_{3,3}$ as a subgraph. So $G_1,G_2 \in R(G,C)$ implies $W(G_1) \neq W(G_2)$.

Sufficiency:

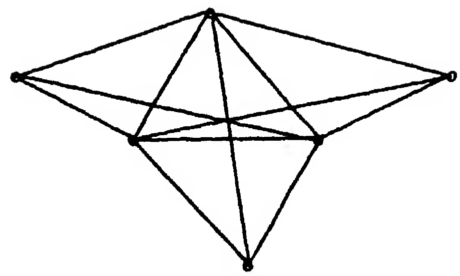
Assume that the conditions of the theorem are satisfied. We claim that G is planar. Since each G_i is planar, $|C| \le 4$. If $|C| \le 3$, then $G=(G_1 \cup G_2 \cup \ldots \cup G_r)= (G_1 \cup G'_2 \cup \ldots \cup G'_r)$, where $G'_i=(G_i-(C-W(G_i))$. Again $|W(G_i)|\le 2$. So by repeated applications of Lemma 2.8.1, G is planar. So let $|C|=4$. Let $C=\{v_1,v_2,v_3,v_4\}$. Wlg, assume that $|R(G,C)|=4$ and $R(G,C)=\{G_1,G_2,G_3,G_4\}$. Assume that $W(G_i)=C-\{v_i\}$, $1\le i\le 4$. Let P_1 be a planar representation of G_1 s.t. the faces f_1,f_2,and f_3 are interior faces, where $\{v_1v_2,v_2v_3,v_3v_1\}$, $\{v_1v_2,v_2v_4,v_4v_1\}$, $\{v_1v_3,v_3v_4,v_4v_1\}$ are the boundaries of f_1,f_2, and f_3, respectively. Let P_2, P_3, and P_3 be some planar representations of G_2-v_2, G_3-v_3, and G_4-v_4 s.t. $\{v_1v_3,v_3v_4,v_4v_1\}$, $\{v_1v_2,v_2v_4,v_4v_1\}$, $\{v_1v_2,v_2v_3,v_3v_1\}$ are the boundaries of the exterior faces of P_2, P_3 and P_4, respectively. Now construct a planar representation P'_1 of G_1 from P_1 by suitably expending f_1, f_2, and f_3 s.t. $\delta(f_1)=\delta(ex(P_4)$, $\delta(f_2)=\delta(ex(P_3)$, and $\delta(f_3)=\delta(ex(P_4)$. Then $(P'_1 \cup P_2 \cup P_3 \cup P_4)$ is a planar representation of $G'=(G_1 \cup G_2 \cup G_3 \cup G_4)$. Now $G=G' \cup (G'_5 \cup G'_6 \cup \ldots \cup G'_r)$, where $G'_i=(G_i-(C-W(G_i))$, $5\le i\le r$. Since $|W(G_i)|\le 2$, by Lemma 2.8.1, G is planar.∎

Theorem 2.8.3: A graph G is chordal planar iff it contains none of the graphs in Figure 2.8.1 as an induced subgraph.

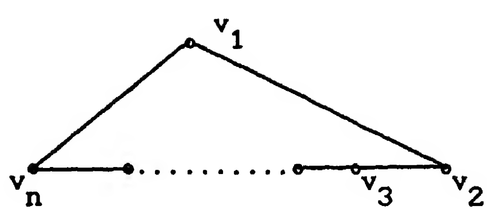
Proof: Clearly, each of the graphs in Figure 2.8.1 is chordal but not planar. So we prove the sufficiency. Let G be a graph s.t.each of its induced subgraphs is chordal planar but G is not. If G is not chordal, by the minimality of G, G will be isomorphic to a C_n, $n\ge 4$, which is D_3. Assume that G is chordal. If G has no separating clique, then G must be isomorphic to a D_1. If G has a separating clique, then as in the proof of Theorem 2.8.2, G will be isomorphic to D_2 of Figure 2.8.1.∎



D_1



D_2



D_3

Figure 2.8.1: Forbidden Subgraphs For Chordal Planar Graphs.

2.9 A Parallel Recognition Algorithm for Chordal Planar Graphs:

In this section we propose an NC algorithm to recognize chordal planar graphs in parallel. This can be done by the following two steps.

Step 1: Test whether the input graph G is chordal.

Step 2: Test whether the graph G is planar.

If the answer to both Step 1 and Step 2 are yes, then G is chordal planar, otherwise not.

The first parallel chordal graph recognition algorithm is due to Edenbrandt[37], which runs in O(log n) time and uses $O(n^3m)$ processors on a CRCW PRAM. Chandrasekharan et al [21] improved Edenbrandt's result by proposing an O(log n) time and $O(n^4)$ processors chordal graph recognition algorithm on a CRCW PRAM. Noar et al [95] then suggested an $O(\log^2 n)$ time and $O(n^3)$ processors CREW PRAM chordal graph recognition algorithm, which can be implemented in O(log n) time and using $O(n^3)$ processors on a CRCW PRAM and hence an improvement over Chandrasekharan et al [21]'s result. P.Klein [80] presented various algorithms including an $O(\log^2 n)$ and O(n+m) chordal graph recognition algorithm. This algorithm is close to optimal by a factor of $\log^2 n$. So Step 1 can be implemented in parallel.

Similarly there are several parallel planarity algorithms (see [75], [81], and [106]). The algorithm due to Ramachandran et al [106] is optimal.

So Chordal planar graphs can be recognized in $O(\log^2 n)$ time using O(n) processors on a CRCW PRAM, as m=O(n) for a planar graph, and the complexity of the chordality testing dominates that of the planarity testing.

In this section we show that chordal planar graphs admit good structural characterization which is suitable for parallel computation.

The following Theorem whose proof follows from Theorem 2.8.3 is the back bone of our chordal planar graph recognition algorithm.

Theorem 2.9.1: A chordal graph G is planar iff (i) $|C|\le 4$ for every clique C of G, and (ii) there exist no separated graphs G_i and G_j w.r.t. any separating clique C of G s.t. $W(G_i)=W(G_j)$, and $|W(G_i)|=3$.

Next we present the Chordal Planarity Test Algorithm.

Algo Chordal Planarity Test:

Input: A Graph G.

Output: "yes" if G is chordal and planar, "no" otherwise.

Method:

Begin

Step 1: Test, in parallel, whether G is chordal;

If G is not chordal then Output "No";

Step 2: Find the set S={ C s.t. C is a maximal clique of G s.t. $|C|\ge 4$}; Let $S=\{C_1,C_2,\ldots,C_k\}$;

Step 3: For all i:=1 to k Do in parallel

Begin

If $|C_i|>4$ Then output "NO";

(* let $C_i=\{v_{i_1},v_{i_2},v_{i_3},v_{i_4}\}$*);

If C_i is not a separating clique Then

B[i]=1

Else

Let $H_1,H_2,\ldots,H_r$ be the connected components of $G-C_i$;

For $1\le j\le k$ and $1\le s\le 4$ Do in parallel

A[j,s]:=0;

For j=1 to 4 Do in parallel

For each $v \in N(v_{i_j})$ Do in parallel

If $v \in V(H_t)$ Then A[t,j]:=1;

For j:=1 to r Do in parallel

$$D[i] := \sum_{j=1}^{4} A[i,j];$$

If either (i) D[i]=3 for at least four i's or

A[i,j]=A[t,j] for $1 \le j \le 4$, and D[i]=3 Then

B[i]:=0

Else B[i]:=1;

End;

Step 4: If $\sum_{i=1}^{k} B[i]=k$ Then output "Yes", otherwise output "No";

END.

Theorem 2.9.2: Algo Chordal Planarity Test is correct and it takes $O(\log^2 n)$ time and $O(n^2+nm)$ processors on a CRCW PRAM.

Proof: The correctness of Algo Chordal Planarity Test follows from Theorem 2.9.1.

Step 1 and Step 2 can be implemented in $O(\log^2 n)$ time and using O(n+m) processors on a CRCW PRAM using P.Klein's Algorithm (see[80]). Since the connectivity and connected components of a disconnected graph can be found out in O(logn) time and O(n+m) processors on a CRCW PRAM (see [121,131]), Step 3 can be implemented in $O(\log^2 n)$ time and using $O(n^2+nm)$ processors, because $k \le n$, for a chordal graph(see [57]). Step 4 can be implemented in O(log n) time and using O(n) processors(see[55]). So our Theorem is proved.∎

Next we show that Planar k-trees can be recognized in $O(\log^2 n)$ time and using $O(n^2)$ processors on a CRCW PRAM.

This can be done by testing whether the given graph is chordal planar and then testing whether it is a k-tree. The only parallel k-trees recognition algorithm is due to Chandrasekharan et al [21] which runs in O(log n) time and uses $O(n^4)$ processors on a CRCW PRAM. However, below we

present a new characterization of k-trees which can be used to recognize k-trees in linear sequential time and in $O(\log n^2)$ parallel time using $O(n+m)$ processors on a CRCW PRAM.

Theorem 2.9.3: A connected chordal graph G is a k-tree iff it has exactly (n-k) maximal cliques each of size (k+1).

Proof: Necessity:

Let G be a k-tree on n vertices, and let $\alpha=(v_1,v_2,\ldots,v_n)$ be a PEO of G. Let $N(v_i,\alpha)=(\{v_j \text{ s.t. } v_iv_j \in E(G) \text{ and } i<j\}\cup\{v_i\})$. It is easy to check that $N(v_1,\alpha)$, $N(v_2,\alpha),\ldots,N(v_{n-k},\alpha)$ are the only maximal cliques of G. Furthermore, each maximal clique is of size (k+1).

Sufficiency:

We prove by induction on n, the number of vertices of G. Since G has a clique of size (k+1), $n \geq k+1$. For $n=k+1$, our Theorem is easily seen to be true. Let G have n vertices. Let v be a simplicial vertex of G. Now $N(v)\cup\{v\}$ is a maximal clique of G. So $\deg(v)=k$. Let $G'=G-v$. Now G' satisfies the assumption of our Theorem. So by induction principle, G' is a k-tree, whence G is a k-tree.█

Since (i) connectedness of a graph can be tested in $O(n+m)$ sequential time[54](in $O(\log n)$ time and using $O(n+m)$ processors on a CRCW PRAM [121,131]), (ii) chordal graphs can be recognized in $O(n+m)$ sequential time[57,114,129], (in $O(\log^2 n)$ time and $O(n+m)$ processors on a CRCW PRAM [80]), (iii) The set of maximal cliques of a chordal graphs can be computed in $O(n+m)$ sequential time [49,57](in $O(\log^2 n)$ time and $O(n+m)$ processors on a CRCW PRAM [80]), and (iv) $\sum(|C|,\ C \in \mathcal{C}(G)) = O(n+m)$ for a chordal graph G, the condition of Theorem 2.9.3 can be tested in $O(n+m)$ sequential time(in $O(\log^2 n)$ time and $O(n+m)$ processors on a CRCW PRAM), whence k-trees can be recognized in $O(n+m)$ sequential time(in $O(\log^2 n)$ time and $O(n+m)$ processors on a CRCW PRAM) using the Theorem 2.9.2. So Planar

k-trees can be recognized in $O(\log^2 n)$ time and $O(n^2)$ processors on a CRCW PRAM, as $m = O(n)$ for planr graph.

2.10 Forbidden Subgraphs For RDV Graphs:

As we have seen in this chapter that to implement our approach to find forbidden subgraphs for RDV graphs, we need the separator Theorem for RDV graphs. Unfortunately, the "if" part of the Theorem 1.3.14, the separator Theorem for RDV graphs due to Monma and Wei[92], is not correct. The incorectness of the "if" part is shown in the following Proposition.

Proposition 2.10.1: The separated graphs G_1, G_2, G_3, and G_4 of G w.r.t. C satisfy the necessary conditions of Theorem 1.3.14(c), but G is not an RDV graph, where G, G_1, G_2, G_3, G_4, and C are as in Figure 2.10.1.

Proof: Color G_1 and G_4 by the color 1, and G_2, and G_3 by color 2. Clearly the separated graphs satisfy all the necessary conditions of Theorem 1.3.14(c).

Next we claim that G is not an RDV graph. Since the tree T in Figure 2.10.1 is the unique UV clique tree for G, $2 \in (C_1 \cap C_2 \cap C_3 \cap C)$, and $5 \in (C \cap C_3 \cap C_4 \cap C_5)$ there is no RDV clique tree for G, because for every RDV clique tree T′ for G, $T'[\{C_1, C, C_3, C_2\}]$ and $T'[\{C_5, C, C_3, C_4\}]$ will be directed paths of T′ and the tree T" obtained from T′ by ignoring the direction is isomorphic to the tree T in Figure 2.10.1. So G is not an RDV graph.∎

We below present the separator Theorem for RDV graphs which is a modification of Theorem 1.3.14(c).

Theorem 2.10.2: G is an RDV graph iff the G_i's can be two-colored such that no antipodal pairs have the same color, and that in one color every subgraph has an RDV clique tree rooted at C, and that in the other color no two subgraphs are unattached, and that no two relevant cliques are

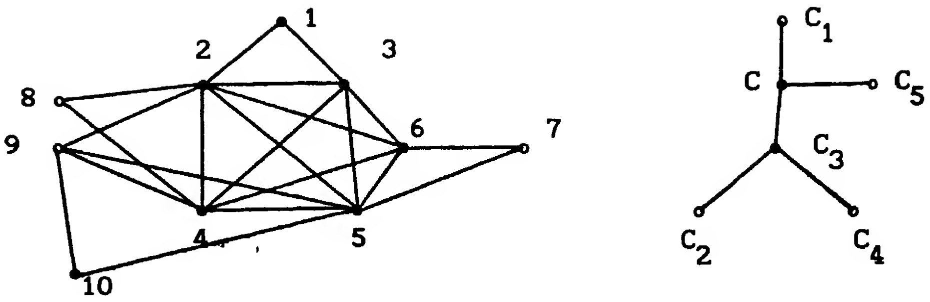
unattached, and every subgraph (with one possible exception) has an RDV clique tree rooted at a relevant clique. The exceptional subgraph, should it exit, is dominated by every other subgraph of the same color, and it has an RDV clique tree in which the vertex C has out degree zero.

Proof: Necessity:

Let T be any RDV clique tree for G. If C is the root of T, then color all the subgraphs with color one and it is easy to see that the G_i's satisfy our Theorem. So assume that C is not the root of T. Color a separated graph by color 1 if it lies in an out going branch w.r.t. C otherwise color it by color 2. Note that antipodal graphs receive different colors in the above coloring.

Let T^* be the subtree of T rooted at C. Then T^* is an RDV clique tree for G^* where $G^* = \cup \{G_i$ s.t. G_i is colored 1}. For every subgraph G_i having color 1, an RDV clique tree T_i rooted at C can be easily constructed from T^*. Next, we consider the graphs having color 2.

The vertices corresponding to the relevant cliques form a contiguous part of the path from the root to C. Hence no two relevant cliques are unattached, and so no two separated graphs are unattached. Let C_i^* be the relevant cliques of G_i that is closest to the root. Let T_i be the subtree of T rooted at C_i^*. From T_i it is easy to construct in the same way as in color 1, an RDV clique tree T_i^* for G_i rooted at C_i^*. The only possible exception is the subgraph containing the root clique, say G_i'. Note that exception occurs exactly when the root clique is not a relevant clique. In this case G_i' is dominated by every other separated graphs having color 2 and the tree T_i' obtained from T by removing T^* is an RDV tree for G_i'. A Clique tree for G_i' with C as a leaf can be easily derived.

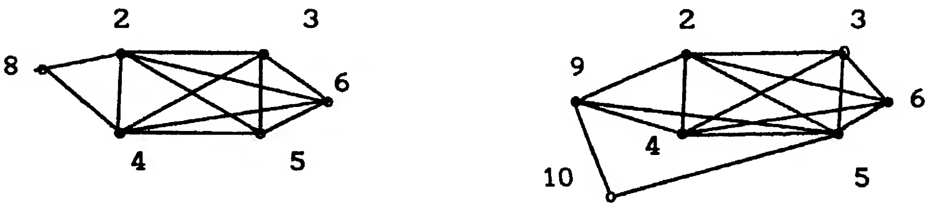


G T

$C=\{2,3,4,5,6\}$, $C_1=\{1,2,3\}$, $C_2=\{8,2,4\}$, $C_3=\{2,4.5.9\}$,

$C_4=\{9,10,5\}$, $C_5=\{5,6,7\}$

G_1 G_4



G_2 G_3

Figure 2.10.1: A Counter Example to Theorem 1.4.14(c).

Sufficiency:

Let the separated graphs be colored in two colors, say color 1 and color 2, satisfying the conditions of our theorem. The RDV clique trees rooted at C for the subgraphs with color 1 can be glued together by the following construction rule R_1 to form an RDV clique tree T' rooted at C for the subgraph G^*, where $G^* = \bigcup_{i \in S_1} G_i$ and S_1={i s.t. G_i is given color 1}.

Rule R_1: We will suggest a recursive construction rule. Since no two separated graphs having color 1 are antipodal, by Proposition 1.3.13, G_i's, $i \in S_1$ can be ordered such that $G_i > G_j$ implies $i< j$. Let G_1, G_2,..., G_r be this ordering.Let T^i be the RDV clique tree for $G_1 \cup G_2 \cup \ldots \cup G_i$ rooted at C. If G_{i+1} is unattached to every G_j, $1\le j \le i$, then merge the root C of the clique tree of G_{i+1} with the root C of T^i to form T^{i+1}. Otherwise let k be the largest index s.t. $G_k \ge G_{i+1}$. Let C_k be the clique of G_k which is farthest from C and which dominates every relevant cliques of G_{i+1}. Now merge the root of the clique tree of G_{i+1} with C_k and call the new vertex C_k. The tree so obtained is T^{i+1}. It is easy to see that T^{i+1} is an RDV clique tree for $G_1 \cup G_2 \cup \ldots \cup G_{i+1}$.

In the RDV clique tree for each subgraph of the color 2, relevant cliques form the path from the root to the clique C. Hence these trees can be glued together by the same method to construct a clique tree T" rooted at a relevant clique. The clique tree of the exceptional subgraph can be glued with T" to form an RDV clique tree T''' in which C is a leaf. Then T' and T''' can be glued at the clique C to obtain an RDV clique tree T for G.∎

Note that the above characterization of RDV graphs is existential in the sense that neither it tells exactly when a non separating clique of an RDV graph G_i is the root of some RDV clique tree for G_i nor it tells the way to obtain the two coloring satisfying the theorem. To find out minimal

forbidden subgraphs using the above theorem the first step would be to know the structure of the separated graphs satisfying the conditions of the above theorem. We present some results along this line.

Let C" be a separating clique of an RDV graph G, and $G_1, G_2, \ldots, G_r$ be the separated graphs of G w.r.t. C". Then we have the following Lemma whose proof follows from Theorem 2.10.2.

Lemma 2.10.3: G has an RDV tree T rooted at C" iff (i) each G_i has an RDV tree rooted at C" and (ii) there exists no antipodal subgraphs w.r.t. C".

Given an RDV graph G and an arbitrary clique C of G, to know whether G has an RDV tree rooted at C, in view of the above lemma it is enough to consider the clique C to be a nonseparating clique of G.

Let C′ be a separating clique and C be a nonseparating clique of G. Let $S_1(G, C') = \{ G_i$ s.t. G_i has an RDV tree rooted at a relevant clique and every relevant clique dominates every relevant clique of G_C, where G_C is the separated graph containing the clique C $\}$. Let $S_2(G,C') = S - \{ S_1(G,C') \cup \{ G_C\}\}$, where S be the set of separated graphs w.r.t. C′. If no confusion arises, we denote $S_i(G, C')$ by S_i, $i = 1, 2$.

Property P: The set of separated graphs of G w.r.t. a separating clique C′ satisfy the property P if there exists no odd length path in $\mathcal{A}(G,C')$ between two vertices of S_2.

In the following Lemma we answer when can an arbitrary non-separating clique C of an RDV graph G be the root of an RDV tree for G.

Lemma 2.10.4: Let G be an RDV graph and C be a non-separating clique of G. Then G has an RDV tree T rooted at C iff for every separating clique C′ of G the separated graphs w.r.t. C′ satisfy the Property P.

Proof: Necessity:

Let C′ be any separating clique of G. Since G is an RDV graph having an RDV tree T rooted at C, the separated graphs are two colored satisfying

the Theorem 2.10.2 and G_C receives the color 2. Since every separated graph in S_2 receives color 1, there does not exist any odd length path in $\mathcal{A}(G,C')$ between any two members of S_2 as $\mathcal{A}(G,C')$ is two colored in such a way that G_C receives the color 2. Hence S satisfies the Property P.

Sufficiency:

Let C′ be any separating clique of G and S be the set of separated graphs. Let H_1, H_2,..., H_r be the connected components of $\mathcal{A}(G,C')$. Note that each H_i is bipartite and hence two colorable. Since there is no odd length path in $\mathcal{A}(G,C')$ between any two members of S_2, $\mathcal{A}(G,C')$ can be two colored in such a way that every G_i in S_2 receives color 1 and G_C receives color 2. Now an RDV tree T rooted at C can be constructed for G following the rule given in the proof of Theorem 2.10.2.∎

The exceptional graph in the Theorem 2.10.2 is required to have an RDV tree with root other than C.

If a non-separating clique C of an RDV graph G is the root of every RDV clique tree for G, then we show below that G admits some nice structure.

Lemma 2.10.5: Let a non-separating clique C of an RDV graph G be the root of every RDV tree T for G, and G be a minimal graph containing C w.r.t. the above property. Then there exists a separating clique C^* of G s.t. G will have exactly three separated graphs, say G_1, G_2, and G_C , where G_C is the separated graph of G containing C, s.t. $G_1|G_2$, $G_1 \Leftrightarrow G_C$, and $G_2 \Leftrightarrow G_C$. **Proof:** Let T be any RDV clique tree of G rooted at C. Let C′ be the vertex of T closest to C s.t. deg(C′) = r, r>2. Clearly C′ is a separating clique of G. Take C^*=C′. We claim that C^* is the required separating clique of G. Let G_C be the separated graphs of G w.r.t. C^* containing C. Since G is a minimal graph, the path from C to C^* is the RDV clique tree for G_C. Again no two separated graphs other than G_C are antipodal. Since G is a minimal graph,

each branch of T at C^* corresponds to exactly one separated graph. Since G_C has an RDV clique tree rooted at C^*, there exists a separated graph, say G_1 antipodal to G_C. Again all the separated graphs except G_C are pair wise unattached. If no G_i other than G_1 is antipodal to G_C, then $G-G_1 \cup \{ C^* \}$ has an RDV clique tree rooted at C^*. Since G_1 has an RDV clique tree rooted at C^*, these trees can be glued to form an RDV clique tree T rooted at C^* for G, which is a contradiction. Hence there exists another separated graph other than G_1, say G_2, which is antipodal to G_C. Then G_C, G_1 and G_2 are the only separated graphs w.r.t. the separating clique C^* as G is a minimal graph. Now $G_1 | G_2$ and $G_i \Leftrightarrow G_C$, $i = 1, 2$. This completes the proof of the Lemma.∎

Using the above mentioned results we can find some forbidden subgraphs for RDV graphs. But we are unable to find all the forbidden subgraphs for RDV graphs, and strongly feel that our results can be used to find all the forbidden subgraphs for RDV graphs.

CHAPTER 3
INTERSECTION GRAPHS OF DISJOINT PATHS IN A TREE

3.1 Introduction:

In this chapter we study the intersection graphs of edge-disjoint paths in a tree, i.e. CV-graphs and the intersection graphs of vertex disjoint paths in a tree, i.e. PV-graphs.

We first present several characterizations of CV-graphs, including the forbidden subgraph characterization. We present a sequential linear time algorithm as well as a parallel NC algorithm for recognizing CV-graphs and for constructing a CV clique tree for a CV-graph. Our parallel algorithm runs in $O(\log^2 n)$ time and uses $O(n+m)$ processors on a CRCW PRAM.

We next show that the characterization of PV-graphs due to samy et al [115] is not correct. We characterize PV-graphs following the framework of Monma and Wei[92], and then find out minimal forbidden subgraphs for this class following the framework introduced in the last chapter. Finally we present a polynomial algorithm for recognizing and for constructing a PV-clique tree for a PV-graph.

3.2 Characterizations of CV-graphs:

Clearly block graphs are a subclass of chordal graphs. Since various subclasses of chordal graphs admit characterizations in terms of intersection graphs of subtrees in a tree with specified properties, it is natural to ask whether block graphs admit such characterization. On the other hand it is natural to ask, looking at the existing literature, what is the class of intersection graphs of edge disjoint paths in a tree. In this section we show that block graphs are exactly the intersection graphs of edge disjoint paths in a tree.

We present below several characterizations of CV graphs.

Theorem 3.2.1: For any graph G the following are equivalent.

(a) G is the intersection graph of a family F of edge disjoint paths in some tree T_1.

(b) G is the intersection graph of a family F of edge disjoint subtrees of some tree T_2.

(c) There exists a tree T with $V(T) = C(G)$ s.t. $F = \{T[C_v(G)],\ v \in V\}$ is an edge disjoint family of paths of T.

(d) G contains neither K_4-e nor C_n, $n \geq 4$ as an induced subgraph.

(e) G is a block graph.

(f) $|C(G)| = |B(G)|$.

A tree T satisfying (c) is called a CV clique tree for G.

Proof: (c) ⇒ (a) and (a) ⇒ (b) are trivial.

(b) ⇒ (e): Since by [67, p 27], any two vertices in a nontrivial block, i.e. a biconnected component, lie on a common cycle, to show the existence of a block graph, it suffices to prove that any two vertices on a cycle are adjacent.

Let $S_1, S_2, \ldots, S_r$ be edge disjoint subtrees of a tree T_2 that form a cycle in G. If every two of them intersect, then by the Helly property of vertex paths[57] they share a common vertex and hence form a clique. Thus suppose some two are disjoint, say S_1 and S_k. Since $S_1, S_2, \ldots, S_r$ are in T_2, there is a unique shortest path from S_1 to S_k in T_2. Let e be any edge on this path. Deletion of e splits the tree into two components with S_1 in one part and S_k in the other. Now $S_{k+1} \cup S_{k+2} \cup \ldots \cup S_r$ is connected and joins S_k to S_1. Hence this union must contain the edge e. Again $S_1 \cup S_2 \cup \ldots \cup S_k$ is connected and contains the edge e. Hence two different subtrees in the cycle must pass through e, a contradiction.

(e) ⇒ (d) : Trivial.

(d) ⇒ (e) : Again we need only to show that any two vertices on a cycle are adjacent. We prove by induction on the length k of the cycle. If $k = 4$, then the cycle contains a chord. Since it cannot be K_4-e, the other chord must also be there as well, so it is complete. Now if $k > 4$, let v and w be any two vertices of the cycle. The cycle has a chord which breaks it into two smaller cycles, which by induction are complete. Thus if v and w are in the same smaller part, they are adjacent. If not, let y and z be the ends of the chord. Then v, w, y, z form a K_4-e unless v and w are adjacent.

(e) ⇒ (c) :

Let G be a block graph. We use induction on k, the number of blocks of G, to prove that G has a CV clique tree T. If $k = 1$, we take $T = K_1$ with $V(T)=\{V(G)\}$. Assume that our claim is true for all block graphs having k or fewer blocks. Let G be a block graph having k+1 blocks. Let C be a boundary block of G, i.e. a block containing exactly one cut vertex of G. Let y be the unique cut vertex of G s.t. $y \in C$. Let $G'=G[V-(C - \{y\})]$. Clearly G′ is a block graph with exactly k blocks. So by induction hypothesis, G′ has a CV clique tree T'. Let $\Pi(y)$ be the path corresponding to y in T' and C′ be an end vertex of $\Pi(y)$. Construct the tree T from T' by taking a new vertex and labeling it by C and making it adjacent to C′ only. We now show that the resulting tree T is a CV clique tree for G. Let $v \in V(G)$. If $v \notin C$, then $\Pi(v)$ is a path in T', so a path in T. If $v \in C -\{y\}$, then $T(C_v)=\{C\}$, which is a path in T. If $v = y$, then $T(C_v)$ is a path in T by the construction of T from T'. Hence $F =\{T(C_v), v \in V\}$ is a family of paths in T. Let $u, v \in V(G)$. Now $|T(C_u) \cap T(C_v)| = 1$ if u and v both belong to C and $|T(C_u) \cap T(C_v)| \leq 1$ otherwise. Hence F is a family of edge disjoint paths in T, whence T is a CV clique tree for G.

(e) ⇒ (f)

This follows from the fact that every maximal clique of G lies in

exactly one block of G and each block of G is complete.

(f) ⇒ (e)

We prove this by induction on k, where $k=|\mathcal{B}(G)|$. If $k=1$, then as $|\mathcal{C}(G)|=|\mathcal{B}(G)|=1$, G is complete, and hence a block graph. Assume that every graph G with $|\mathcal{C}(G)|=|\mathcal{B}(G)|$, $|\mathcal{B}(G)| < k$ is a block graph. Let G be s.t. $|\mathcal{C}(G)|=|\mathcal{B}(G)|$, and $|\mathcal{B}(G)|=k$. Let v be a cut vertex of G, and $G_i=G[\{V_i \cup \{v\}\}]$, where $H_i(V_i,E_i)$, $1 \le i \le r$, $r \ge 2$ be the connected components of G. For every graph H, we have $|\mathcal{C}(H)| \ge |\mathcal{B}(H)|$. Therefore, $|\mathcal{C}(G_i)| \ge |\mathcal{B}(G_i)|$, $1 \le i \le r$. Since $|\mathcal{C}(G)|=\sum_{i=1}^{r}|\mathcal{C}(G_i)|=|\mathcal{B}(G)| = \sum_{i=1}^{r}|\mathcal{B}(G_i)|$, $|\mathcal{C}(G_i)| = |\mathcal{B}(G_i)|$, $1 \le i \le r$. So by induction hypothesis each G_i is a block graph. Since $V(G_i) \cap V(G_j)= \{v\}$, $1 \le i \ne j \le r$, G is a block graph.∎

Let G be a CV graph and C be a separating clique of G. Let $G_i= G[V_i \cup C]$, $1 \le i \le r$, $r \ge 2$, be the separated subgraphs.

Proposition 3.2.2: $W(G_i)$ is a singleton set, $1 \le i \le r$.

Proof: Clearly, $W(G_i)$ is non empty. If $|W(G_i)| \ge 2$, then it is easy to show that G_i contains $K_4 - e$ as an induced subgraph, a contradiction to the fact that G is a block graph.∎

Now the characterization of CV graphs due to Samy et al [34] becomes a corollary to our Theorem 3.2.1.

Corollary 3.2.3: G is a CV graph iff each separated subgraph is a CV graph.

Proof: Necessity is trivial. For sufficiency, let each separated subgraph G_i, $1 \le i \le r$, be a CV graph. So by Theorem 3.2.1, each G_i is a block graph. Again by Proposition 3.2.2, each $W(G_i)$ is a singleton set. Hence G is a block graph. So by Theorem 3.2.1, G is a CV graph.∎

3.3 Recognition Algorithm for CV-graphs:

We now suggest a linear time algorithm to recognize a CV graph and to construct an intersection model if the graph is a CV graph.

Algorithm Test:

INPUT: A graph $G = (V,E)$ in adjacency list representation.

OUTPUT: 'No' if G is not a CV graph. Otherwise a CV clique tree $\mathcal{T}$ for G.

METHOD:

BEGIN

STEP 1 : If G is not a block graph, then output 'NO'.

STEP 2 : Find all cliques of G. Let $C_1, C_2, \ldots, C_r$ be the cliques of G.

STEP 3 : Find the set $\{ C_{i_1}, C_{i_2}, \ldots, C_{i_{r_i}} \}$ of cliques containing v_i, $1 \le i \le n$.

STEP 4: $\mathcal{T} := \mathcal{T}(V_0, E_0)$, where V_0 is the set of all cliques of G and $E_0 = \phi$.

STEP 5: For $i := 1$ to n do

$\mathcal{T} := \mathcal{T}(V_i, E_i)$, where $V_i = V_0$ and

$E_i = E_{i-1} \cup \{ G_{i_j} G_{i_{j+1}}, 1 \le j \le r_i - 1 \}$ if $r_i > 1$ else $E_i :=$

E_{i-1}.

END.

Theorem 3.3.1: Algorithm Test is correct.

Proof: The correctness of recognition part follows from Theorem 3.2.1. We claim that the graph $\mathcal{T}$ constructed by the Algorithm Test is a CV clique tree for G, if G is a CV graph.

If G is a block graph, then a vertex is a cut vertex iff it lies in more than one clique. Again it is an easy exercise to show that the number of blocks of a connected graph G is equal to $1 + \sum_{v \in V} (b(v) - 1)$, where $b(v)$ is the number of blocks of G containing v. Hence T has $(|\mathcal{C}(G)| - 1)$ edges. Now $\mathcal{T}$ is acyclic since $| \mathcal{C}_v(G) \cap \mathcal{C}_w(G) | \le 1$ for a block graph G. Hence $\mathcal{T}$ is a tree. Again $\mathcal{F} = \{ \mathcal{T}[\mathcal{C}_v(G)], v \in V \}$ is a family of paths by the

construction of $\mathcal{T}$. By Theorem 3.2.1, G is a block graph. Let x and y be in V. If x and y belong to the same block of G, then $|\mathcal{C}_x(G) \cap \mathcal{C}_y(G)| = 1$, Otherwise $|\mathcal{C}_x(G) \cap \mathcal{C}_y(G)| = 0$. Therefore, $|\mathcal{T}[\mathcal{C}_x(G)] \cap \mathcal{T}[\mathcal{C}_y(G)]| \le 1$ and hence, F is a family of edge disjoint paths in $\mathcal{T}$. Thus, $\mathcal{T}$ is a CV clique tree for G. So Algorithm Test is correct.▌

The following shows that Algorithm Test runs in O(n+m) time.

Theorem 3.3.2: Algorithm Test runs in O(n+m) time.

Proof: For step 1, first we test whether G is chordal. If G is not chordal, then G is not a block graph. Then we find the biconnected components of G. By Theorem 3.2.1 (f), G is a block graph iff $|\mathcal{C}(G)|=|\mathcal{B}(G)|$. Since, as mentioned earlier, chordal graphs can be recognized in linear time[57,114,129] and maximal cliques of a chordal graphs can be computed in O(n+m) time[49], Step 1 takes O(n+m) time. Since $\sum_{i=1}^{r} |C_i| = O(m+n)$ and $r = O(n)$, other steps of Algorithm Test can be implemented in linear time. So Algorithm Test takes O(n+m) time.▌

In view of the above discussion we have the following theorem.

Theorem 3.3.3: CV graphs can be recognized in linear time; and for a CV graph a CV clique tree can be constructed in linear time.

We now show how Algorithm Test can be implemented in $O(\log^2 n)$ time and in O(n+m) processors on a CRCW PRAM.

The recognition of chordal graphs and finding the maximal clique of chordal graphs can be done in $O(\log^2 n)$ time and in O(n+m) processors[80]. Again biconnected components of a general graph can be found out in O(logn) time and in O(n+m) processors [121,131]. So Algorithm Test can be implemented in $O(\log^2 n)$ time and in O(n+m) processors, as in Step 1 of Theorem 3.3.2. Sridhar et al [124] also presents a parallel recognition algorithm of block graphs in same time and processor bound. Now Step 2 can also be done $O(\log^2 n)$ time and in O(n+m) processors[80]. Since $\sum_{i=1}^{r} |C_i| =$

O(m+n), using the sorting algorithm of Cole[28], step 3 can be implemented in O(log n) time and in O(n+m) processors. Let S_i be the set of cliques containing v_i. For each S_i compute a path P_i using the cliques of S_i. Define $\mathcal{T}$ as $V(\mathcal{T}) = \mathcal{C}(G)$ and $E(\mathcal{T}) = \cup P_i$. Since $\sum_{i=1}^{n} |S_i| = O(n)$, this takes O(n) processors and no more than $O(\log^2 n)$ time. So we have the following Theorem.

Theorem 3.3.4: CV graphs can be recognized and for a CV graph an intersection model can be constructed in $O(\log^2 n)$ time and in O(n+m) processors on a CRCW PRAM.

3.4 Characterization of PV-Graphs:

In this section we show that the characterization of PV-graphs due to Samy et al [115] is not correct. We then present a characterization of PV-graphs following the framework of Monma and Wei[92].

First we characterizes PV-graphs in terms of clique trees.

Theorem 3.4.1: A graph G is a PV graph iff there exists a tree T with V(T) = $\mathcal{C}(G)$ s.t. F $=\{T[\mathcal{C}_v(G)],\ v \in V(G)\}$ is a family of vertex disjoint paths in T.

We call a tree satisfying Theorem 3.4.1 a PV clique tree for G. The proof of the above theorem goes in the same line as that of [92,Theorem 1 (b)].

Proof of Theorem 3.4.1:

Given a tree T and a family F of vertex disjoint paths in T, it can be easily seen that G is the intersection graph of F, and hence a PV graph. Conversely, let G be a PV graph, and let (T,F) be a PV representation for G, where T has the smallest possible number of vertices. Every vertex of T corresponds to a set $S \subseteq V$ s.t. G[S] is a complete subgraph of G. We claim that there is a one to one correspondence between the vertices of T and the

cliques of G. Now by Helly property of vertex paths [57], there is a vertex in T corresponding to every clique in G. Suppose there are two distinct vertices v_1 and v_2 in T which correspond in G to S_1 and S_2, respectively, where $S_1 \subset S_2$. Let v_3 be the vertex next to v_1 in the v_1--v_2 path in T, and S_3 be the vertex set of V corresponding to v_3. It is easy to see that $S_1 \subset S_3$. Construct a new tree T' from T by coalescing v_1 and v_3 and eliminating the edge between them. Any path in F that contains v_1 necessarily contains v_3. Therefore, T' is a PV clique tree for G with less number of vertices than that of T, which is contrary to our assumption on T. Therefore the vertices of T correspond to distinct cliques of G, and T is a clique tree for G.∎

Since every PV graph is chordal, throughout this section we let G be a chordal graph, C be a separating clique of G, and $G_i = G[V_i \cup C]$, $1 \le i \le r$, $r \ge 2$ be the separated subgraphs.

Samy et al [115] introduced the notion of PV-graphs. There, they proved the following results.

Proposition 3.4.2: [115] If G is a PV graph, then

(a) The intersection of any three cliques of G is at most a singleton set.

(b) There does not exist more than one pair of antipodal subgraphs w.r.t. any separating clique C of G.

Theorem 3.4.3: [115] (Separator theorem for PV graphs)

Let C be a separating clique of G and G_1, $G_2, \ldots, G_r$, $r \ge 2$ be the separated graphs of G w.r.t. C. Then G is a PV graph iff

(a) Each G_i is a PV graph,

(b) There does not exist more than one pair of antipodal subgraphs w.r.t. C, and

(c) If G_i dominates G_j, then $W(G_j)$ is a singleton set and there is exactly one relevant clique in G_i intersecting $W(G_i) \setminus W(G_j)$ and there does not

exist a pair of antipodal cliques w.r.t. any relevant clique of G_1.

The following Proposition shows that Theorem 3.4.3 is not correct.

Proposition 3.4.4: The graph G of Figure 3.4.1 is a PV-graph, but the separated graphs G_1 and G_2 of G w.r.t. C given in Figure 3.4.1 do not satisfy Theorem 3.4.3 (c). Moreover, the graph H_8 of Figure 3.5.4 is not a PV-graph, but the separated graphs of H_8 w.r.t. any separating clique of H_8 satisfy all the conditions of Theorem 3.4.3.

Proof:

Clearly, G is a PV-graph as it is the intersection graph of the following collection of pair wise vertex disjoint paths in T:

$\{ C_0, C, C_0\text{-}C\text{-}C_1, C\text{-}C_1\text{-}\dots\text{-}C_n, C_1\text{-}C_2, C_2\text{-}C_3, \dots, C_{n-1}\text{-}C_n, C_n \}$. Now $G_1 > G_2$ and each of the cliques $C_1, C_2, \dots, C_n$ intersects v and $v \in W(G_1)-W(G_2)$. So Theorem 3.4.3(c) is not true. Note also that the separated graphs of the graph H_8 in Figure 3.5.4 w.r.t. any separating clique satisfy all the conditions of Theorem 4.3.3, but H_8 is not a PV-graph (This will be shown in Section 3.5).∎

Next we preset some concepts and prove some results on PV-graphs.

Let G be a PV-graph and T be a PV-clique tree for G. For $v \in V$, let $\pi(v)$ be the path corresponding to v in T. Define a function I_T from C(G) to $\{0,1\}$ by $I_T(C) = 1$ if C is an internal vertex of a path $\pi(v)$ for some $v \in V(G)$. Otherwise $I_T(C)=0$.

As the proposition 3.4.2 will be used in our main result, we supply its proof for the sake of completeness.

Proof of Proposition 3.4.2:

(a) Let T be a PV-clique tree for G. Assume that there are three cliques C_1, C_2, and C_3 s.t. $|C_1 \cap C_2 \cap C_3| \geq 2$. Let $\{v,w\} \subseteq C_1 \cap C_2 \cap C_3$. Since $\pi(v)$ and $\pi(w)$ are paths in T containing three common vertices C_1, C_2, and C_3, one of C_1, C_2, and C_3 will be an internal vertex of both the

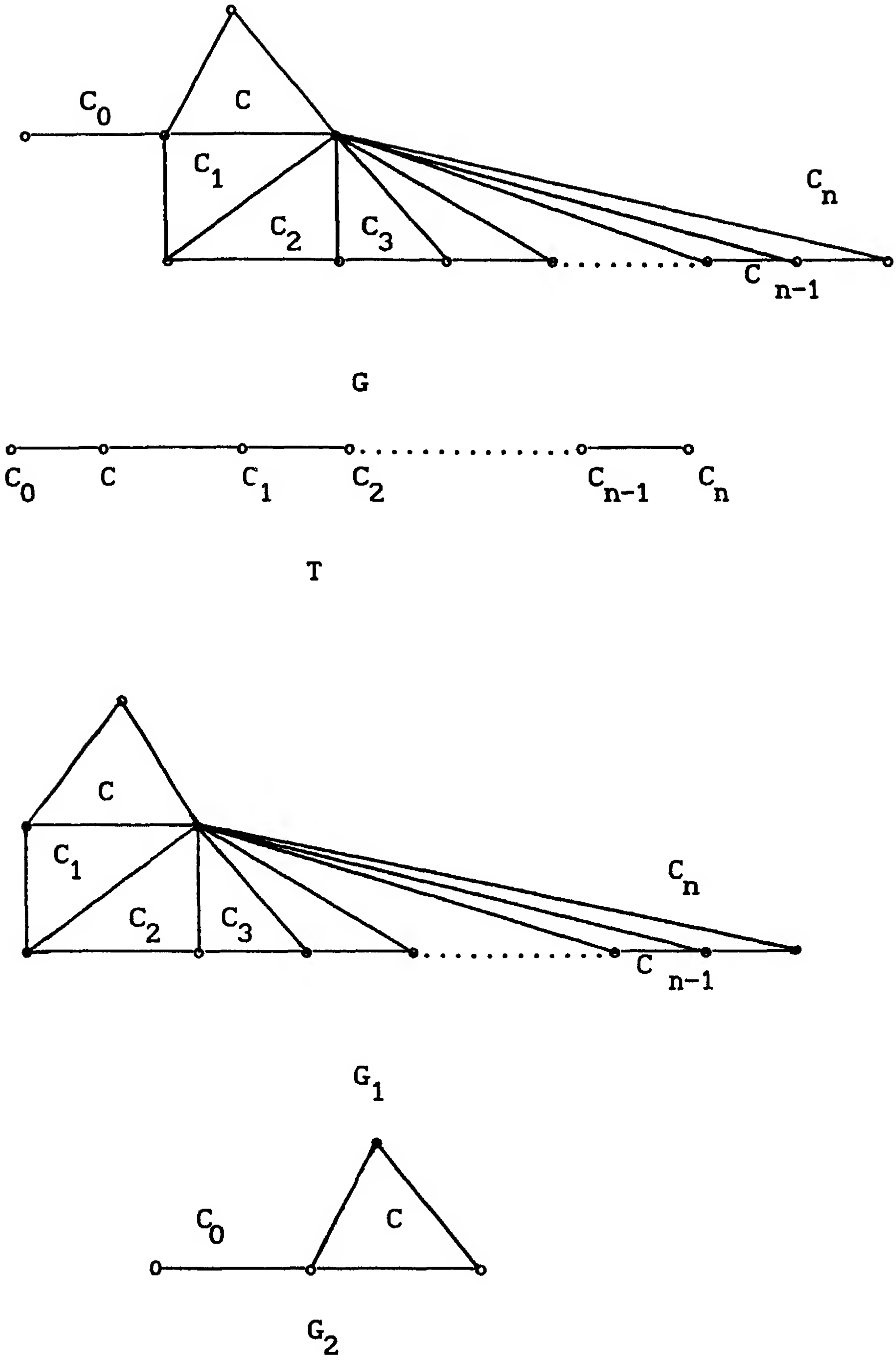


Figure 3.4.1: A PV-Graph G, a PV-clique Tree T and the two Separated Graphs G_1 and G_2 w.r.t. C.

paths, thus contradicting to the fact that T is a PV-clique tree for G.▮

Before proving Proposition 3.4.2(b), we prove certain results on PV-graphs.

Proposition 3.4.5: Let G be a PV-graph. Then G has a pair (G_1,G_2) of antipodal subgraphs w.r.t. a separating clique C iff there exist C_1 in G_1 and C_2 in G_2 s.t. $C_1 \Leftrightarrow C_2$ w.r.t. C.

Proof: Sufficiency follows from the definition of antipodal subgraphs.

Necessity: Since G_1 is antipodal to G_2, $|W(G_i)| \geq 2$ for $i = 1,2$. Let C_i be some principal clique of G_i, $1 \leq i \leq 2$. Now $|C_1 \cap C_2| \geq 1$. By Proposition 3.4.2(a), $|C_1 \cap C_2 \cap C| \leq 1$. So $|C \cap C_1 \cap C_2| = 1$. Hence $C_1 \Leftrightarrow C_2$.▮

Proposition 3.4.6: Let G be a PV-graph. If $G_1 \Leftrightarrow G_2$ w.r.t. C, then the subtrees corresponding to G_1 and G_2 lie in different branches of C in any PV-clique tree T for G.

Proof: Assume that there is a PV-clique tree T for G s.t. the subtrees corresponding to G_1 and G_2 lie in same branch of C. Since $G_1 \Leftrightarrow G_2$, there exist, by Proposition 4.3.5, C_1 in G_1 and C_2 in G_2 s.t. $C_1 \Leftrightarrow C_2$. Let $v \in C_1 \cap C_2 \cap C$. Now the path $\pi(v)$ in T contains C, C_1, and C_2. Wlg, let C_1 lie in the unique path from C to C_2 in T. Let $w \in (C_2 \cap C) - C_1$. Now $\pi(w)$ contains C and C_2 but not C_1, which is impossible because the unique path from C to C_2 passes through C_1. Hence a contradiction arises.▮

Next we prove Proposition 3.4.2(b).

Proof of Proposition 3.4.2(b):

Let $G_1 \Leftrightarrow G_2$ and $G_3 \Leftrightarrow G_4$. Let T be a PV-clique tree for G.

Case I: There exist i and j s.t. $G_i = G_j$, $1 \leq i < j \leq 4$.

Wlg let $G_2 = G_4$. So $G_1 \Leftrightarrow G_2$ and $G_2 \Leftrightarrow G_3$. So by Proposition 3.4.5 there exists $C_i \in G_i$, $1 \leq i \leq 3$, s.t. $C_1 \Leftrightarrow C_2$ and $C_2 \Leftrightarrow C_3$. If $C_3 | C_1$, then let $x \in C_1 \cap C_2 \cap C_3$, and $y \in C_2 \cap C_3 \cap C$. Since $C_3 | C_1$, $x \neq y$. Now, by Proposition 3.4.6, the subtrees corresponding to G_1 and G_2 lie in different branches of C in

In the tree T, let $\pi(C_i,C_j)$ denote the path from C_i to C_j. Observe that C_i and C_j are on the same branch(w.r.t. root C) iff $C \notin \pi(C_i,C_j)$.

Proposition 3.4.8: No G_i contains antipodal cliques(w.r.t.C).

Proof: By proposition 3.4.7, each G_i is a PV-graph which has a clique tree T_i with C as a leaf. Suppose that a subgraph G_i contains cliques C_1 and C_2, where $C_1 \Leftrightarrow C_2$. Let $x \in (C_1 \setminus C_2) \cap C$, $y \in (C_2 \setminus C_1) \cap C$, and $z \in (C_1 \cap C_2 \cap C)$. The path $\pi(z)$ in T_i contains the cliques C, C_1, and C_2. If C_1 lies in between C and C_2 in the path $\pi(z)$, then the path $\pi(y)$ contains both C and C_2, and hence, must contain C_1. But $y \notin C_1$, which is a contradiction. If C_2 lies in between C and C_1 in the path $\pi(z)$, then the path $\pi(x)$ produces a similar contradiction.█

Proposition 3.4.9: Let T be any PV clique tree for G. Let C′ and C" be two cliques on the same branch of C. If C′ and C" are attached, then either $C' \in \pi(C,C")$ or $C" \in \pi(C,C')$. Furthermore, If $C'>C''$, then $C' \in \pi(C, C'')$.

Proof: Let $x \in C' \cap C" \cap C$. Since C′ and C" lie in the same branch of C, either $C' \in \pi(C,C")$ or $C" \in \pi(C,C')$. Now let $C'>C"$. If $C" \in \pi(C,C')$, then $C" \in \pi(y)$ for $y \in ((C' \setminus C") \cap C)$. So $C'>C"$ implies $C' \in \pi(C,C")$.█

Proposition 3.4.10: If $C' \in \pi(C,C")$, then $C' \geq C"$.

Proof: For every $x \in C \cap C"$, we have $C' \in \pi(C.C") \subseteq \pi(x)$, hence $x \in C'$. So $C' \geq C"$.█

Proposition 3.4.11: Let C′ and C" be two relevant cliques of a subgraph G_i, then every clique in $\pi(C',C")$ is in G_i.

Proof: Let $x \in C' \setminus C$ and $y \in C" \setminus C$. Then there is a path P (consisting of vertices) in $G[V_i]$ connecting x and y. Let $\pi(P)$ be the union of all paths (consisting of edges) in T corresponding to the members of P. So $\pi(C',C") \subseteq \pi(P)$. Every clique in $\pi(C',C")$ contains some vertex of V_i, hence is in G_i, because vertices from different subgraphs are nonadjacent.█

A separated graph G is said to be incompatible w.r.t. a vertex v ∈

$W(G_i)$ if for every PV-clique tree T_i for G_i, $I_{T_i}(C_i) = 1$, where $C_i \neq C$ is an end vertex of the path $\pi(v)$ in T_i. A pair (G_i,G_j) is said to be an incompatible pair if $|W(G_i) \cap W(G_j)|=1$, and either (1) $G_i > G_j$ and G_i is incompatible w.r.t. v_j, or (2) $G_i \sim G_j$ and G_i and G_j are incompatible w.r.t. v_j, where $\{v_j\}=(W(G_i) \cap W(G_j))$.

Proposition 3.4.12: Let G be a PV-graph. Then the subtrees corresponding to an incompatible pair (G_i,G_j) lie in different branches of C in any PV-clique tree T for G.

Proof: Let T be a PV-clique tree for G and (G_i,G_j) be an incompatible pair w.r.t. C. Assume that the subtrees corresponding to G_i and G_j lie in the same branch of C in T. Construct T_i and T_j from T as in Proposition 3.4.7. Let $\{v_j\}= W(G_i) \cap W(G_j)$. We consider two cases separately.

Case 1: $G_i > G_j$.

Let $C_i \neq C$ be an end vertex of $\pi(v_j)$ in T_i. Since (G_i,G_j) is an incompatible pair, $I_{T_i}(C_i)=1$. Since, by proposition 3.4.9, $C_i \in \pi(C,C_j)$, where C_j is a relevant clique of G_j, C_i is an internal vertex of the path $\pi(v_j)$ in T. So $I_T(C_i) \geq 2$. But this contradicts the fact that T is a PV -clique tree for G.

Case 2: $G_i \sim G_j$.

By Proposition 3.4.9, either $C' \in \pi(C,C'')$ or $C'' \in \pi(C,C')$, where C' and C'' are any principal clique of G_i and G_j, respectively. Wlg, let $C' \in \pi(C,C'')$. Then, using the similar argument as in case 1, we can show that $I_T(C_i) \geq 2$, where C_i is as in case 1. Hence the Proposition is proved.▮

For any separated graph G_i, let $D(G_i)= \{G_j \text{ s.t. } G_i > G_j\}$. Define a relation R on $D(G_i)$ by $G_j \; R \; G'_j$, for G_j, $G'_j \in D(G_i)$, iff $G_j \sim G'_j$. Note that R is an equivalence relation on $D(G_i)$. Let $N(D(G_i))$ denote the number of equivalence classes of $D(G_i)$ under the relation R.

Next We present some necessary conditions for a PV-graph G.

Proposition 3.4.13: For a PV graph G, the following conditions hold:

(1) For every G_i, $N(D(G_i)) \leq 2$.

(2) $N(D(G_i)) = 2$ for at most one separated graph G_i.

(3) Let $G_i > G_j$. Then $W(G_j)$ is a singleton set, and there is no incompatible pair (G'_i, G'_j) s.t. $G'_i, G'_j \in D(G_i)$ and $W(G'_i) \neq W(G_j)$.

(4) If $N(D(G_i)) = 2$, then there does not exist an incompatible pair (G'_i, G'_j) of separated graphs.

Proof: (1) Suppose there exists some G_i s.t. $N(D(G_i)) \geq 3$. Let $G_j, G'_j, G''_j \in D(G_i)$ be s.t. $\{v_j\}=W(G_j)$, $\{v'_j\} = W(G'_j)$, and $\{v''_j\} = W(G''_j)$. So v_j, v'_j, and v''_j are all distinct. Since C and C_i lie on each of the paths $\pi(v_j)$, $\pi(v'_j)$, and $\pi(v''_j)$ of length two or more, either C or C_i will be an internal vertex of at least two of them.

(2) Suppose there exist G_i s.t. $N(D(G_i)) = 2$, $1 \leq i \leq 2$. Let $G'_1, G_1'' \in D(G_1)$ and $G'_2, G_2'' \in D(G_2)$ be s.t. $v_1 \neq v_2$ and $v_3 \neq v_4$, where $\{v_1\} = W(G'_1)$, $\{v_2\} = W(G''_1)$, $\{v_3\} = W(G'_2)$, and $\{v_4\} = w(G''_2)$. We consider two cases separately.

Case 1: There exist i and j, $1 \leq i < j \leq 4$, s.t. $v_i = v_j$.

Wlg, let $v_2 = v_3$. Let C_1 and C_2 be some principal cliques of G_1 and G_2 respectively. Now $|W(G_1) \cap W(G_2)| \leq 1$. Otherwise, the intersection of C, C_1, and C_2 will contain at least two vertices, a contradiction to Proposition 3.4.2(a). Since $v_2 = v_3$, $G_1 \Leftrightarrow G_2$. Let T be a PV-clique tree for G. So C will be an internal vertex of the path $\pi(v_2)$. Again C_1 or C_2 will be an internal vertex of the path $\pi(v_2)$. Wlg, let C_1 be an internal vertex of the path $\pi(v_2)$. Now either C or C_1 will be an internal vertex of the path $\pi(v_1)$, since $\pi(v_1)$ contains C, C_1, and C'_1, where C'_1 is a principal clique of G'_1, and $C_1 > C'_1$. So either $I_T(C) \geq 2$ or $I_T(C_1) \geq 2$, a contradiction to the fact that T is a PV-clique tree for G.

Case 2: v_1, v_2, v_3, and v_4 are all distinct.

Let T be a PV-clique tree for G. Then $\pi(v_1)$ and $\pi(v_2)$ will contain C

and C_1, where C_1 is a principal clique of G_1. Since each of $\pi(v_1)$ and $\pi(v_2)$ contains at least three vertices of T, C will be an internal vertex of either $\pi(v_1)$ or $\pi(v_2)$. By a similar argument C will be an internal vertex of either $\pi(v_3)$ or $\pi(v_4)$. So $I_T(C) \geq 2$, a contradiction.

(3) Let $G_i > G_j$. Let C_i and C_j be some principal cliques of G_i and G_j, respectively. If $W(G_j)$ is not a singleton set, then the intersection of C, C_i, and C_j will contain at least three vertices, a contradiction to Proposition 3.4.2(a). Assume that there exists an incompatible pair (G'_i, G'_j) s.t. G'_i, $G'_j \in D(G_i)$, and $v'_i \neq v_j$, where $\{v'_i\} = W(G'_i)$ and $\{ v_j\} = W(G_j\}$. Since by Proposition 3.4.12, the subtrees corresponding to an incompatible pair lie in different branches of C, C will be an internal vertex of $\pi(v'_i)$. Since, $C_i > C'_i$ and $C_i > C'_j$, where C'_i and C'_j are principal cliques of G'_i and G'_j, respectively, C_i is an internal vertex of $\pi(v'_i)$. Since $C_i > C_j$, either C or C_i will be an internal vertex of $\pi(v_j)$. Then either $I_T(C) \geq 2$ or $I_T(C_i) \geq 2$, which is not true for a PV-clique tree T for G.

(4) Let T be a PV-clique tree for G. Let $G_1 > G_2$, $G_1 > G_3$, and $x_2 \neq x_3$, where $\{x_2\}=W(G_2)$, and $\{x_3\}=W(G_3)$. Assume that (G_4,G_5) is an incompatible pair and $\{x_4\}=W(G_4) \cap W(G_5)$.

Case 1: $x_4 \neq x_2$, and $x_4 \neq x_3$.

Then by Proposition 3.4.12, C will be an internal vertex of $\pi(x_4)$. Again C will be an internal vertex of either $\pi(x_2)$ or $\pi(x_3)$. So a contradiction arises.

Case 2: Either $x_4 = x_2$ or $x_4 = x_3$.

Wlg, $x_4=x_2$. If $G_4 \sim G_5$, then this case reduces to Proposition 3.4.13(3). Assume that $G_4 > G_5$. Then, by Propositions 3.4.12 and 3.4.9, C and C_1 will be internal vertices of $\pi(x_4)$, where C_1 is a principal clique of G_1. Again either C or C_1 will be an internal vertex of $\pi(x_3)$, a contradiction.∎

Lemma 3.4.14: A collection of pair wise non-antipodal, pair wise non-incompatible separated subgraphs can be arranged in such a way that either $G_i > G_j$, or $G_i \sim G_j$ and G_j is incompatible w.r.t. x_j, where $\{x_j\} = W(G_j)$, then $i < j$.

Proof: If there are no $G_i \sim G_j, i \neq j$, then by Proposition 1.4.13, we are through. If there are congruent subgraphs $G_i \sim G_j$, We take one subgraph from each congruence class, arrange them using Proposition 1.4.13 and obtain a sequence α. Since there is no incompatible pair of separated graphs, there is at most one graph G_i in a congruence class such that G_i is incompatible w.r.t. x_i, where $\{x_i\} = W(G_i)$. So each congruence class can be arranged as stated in the lemma. We replace each element G_i' of the sequence α by the sequence of the congruence class corresponding to G_i' and obtain the sequence β, which is a desired arrangement.▌

Now we characterize PV-graphs in terms of separated subgraphs.

Theorem 3.4.15: (Separator Theorem for PV graphs)

Let G be a chordal graph and $G_1, G_2, \ldots, G_r$, $r \geq 2$ be the separated graphs of G w.r.t. a separating clique C. Then G is a PV graph iff

(i) Each G_i is a PV graph,

(ii) The intersection of any three cliques of G is at most a singleton set,

(iii) There exists at most one pair of antipodal subgraphs w.r.t. C,

(iv) All the conditions of Proposition 4.3.13 are true,

(v) If there exists a pair (G_1, G_2) of antipodal subgraphs w.r.t. C relevant to a vertex v, then the following conditions hold:

(a) There exists no incompatible pair (G_i, G_j) s.t. $W(G_j) \neq \{v\}$.

(b) There exist no G_i, G_j and G_k s.t. $G_i > G_j$ and $G_i > G_k$ s.t. v, v_j, and v_k are all distinct, where $\{v_j\} = W(G_j)$ and $\{v_k\} = W(G_k)$, and

(vi) There do not exist two pairs of incompatible pair of separated graphs.

Proof: **Necessity:** (i) follows from Proposition 3.4.7.

(ii) and (iii) follow from Proposition 3.4.2.

(iv) follows from Proposition 3.4.13.

(v) Assume that there exists a pair (G_1, G_2) of antipodal separated graphs w.r.t. C relevant to v. Let T be a PV-clique tree for G. By Proposition 3.4.6, the subtrees corresponding to $G_1$and G_2 lie in different branches of C in T. So C is an internal vertex of the path $\pi(v)$ in T. If (a) is not true, then by Proposition 3.4.12, C will be an internal vertex of the path $\pi(v_j)$, where $\{v_j\}=W(G_j)$. So $I_T(C)\geq 2$, which is a contradiction. Assume that (b) is not true. Let C_i,C_j, and C'_j be some principal cliques of G_i, G_j and G'_j, respectively. Since $C_i>C_j$ and $C_i>C'_j$, C_i will be an internal vertex of one of the paths $\pi(v_j)$ and $\pi(v'_j)$, say of $\pi(v_i)$. Again C_i or C will be an internal vertex of the path $\pi(v'_j)$. So either $I_T(C) \geq 2$ or $I_T(C_i) \geq 2$, which is a contradiction.

(vi) Assume that (vi) is not true. Let T be a PV-clique tree for G. By proposition 3.4.12, the subtrees corresponding to an incompatible pair (G_i, G_j) lie in different branches of C in T. If there exist G_i,G_j, and G_k s.t. they are pair wise congruent, and pair wise incompatible, then $\pi(v)$ is not a path in T, where $v \in W(G_i)$; a contradiction to the fact that G is a PV-graph. In other cases it can be seen easily that C will be an internal vertex of more than one path; a contradiction.

Sufficiency:

Let X = $\{G_1,G_2,G_3,\ldots,G_r\}$ be the separated subgraphs. By assumption there is at most one pair of antipodal subgraphs.

Case 1: There is one antipodal pair, say (G_1 , G_2).

Let $Y=\{G_j$ s.t. $G_2 \geq G_j\}$ and Z=X -Y. Now Y and Z are collection of separated graphs satisfying the assumption of Lemma 3.4.14. Let $\{G'_1$, G'_2, $G'_3,\ldots,G'_{t_1}\}$ and $\{G'_1$, G'_2, $G'_3,\ldots,G''_{t_2}\}$ be arrangements of Z and Y according to lemma 3.4.14. We now give a method to construct a tree T_1 for the

collection Z and a tree $T_2$for the collection Y and combine these two trees suitably to obtain a PV clique tree T for G .

Let T'_i be a PV clique tree for G'_i, $1 \le i \le t_1$ s.t. if (i) either $G'_i > G'_j$, or $G'_i \sim G'_j$ and $i < j$, and (ii) if C_i is an end vertex of the path $\pi(v_j)$, where $\{v_j\} = W(G'_i) \cap W(G'_j)$, then $I_{T'_i}(C_i) = 0$. We construct T_1 iteratively . Let $T_1^{(k-1)}$ be the tree obtained from T'_1, $T'_2, \dots, T'_{k-1}$, $k \le t$. We construct $T_1^{(k)}$ as follows :

If $|W(G'_k)| \ge 2$, then merge the leaf vertex C of T'_k with the vertex C of $T_1^{(k-1)}$ to obtain $T_1^{(k)}$. Let $W_k(G') = \{ v'_k\}$.

Subcase 1(a): G'_k is not dominated by any one of G'_1, $G'_2, \dots$, G'_{k-1}.

Then merge the leaf vertex C of T'_k with the vertex C of $T_1^{(k-1)}$to obtain $T_1^{(k)}$.

Subcase 1(b): G'_k is dominated by some G'_j, $j < k$.

Let C′ be an end vertex in the path $\pi(v'_k)$ in the tree $T_1^{(k-1)}$, s.t. $C' \ne C$. Merge the leaf vertex C of T'_k with the vertex C′ of $T_1^{(k-1)}$, call the new vertex C′ and the resulting tree T_1.

For the collection Y, Subcase 1(a) will not occur. Construct a tree T_2 for $\bigcup_{i=2}^{t_2} G''_i$ using the similar procedure as above.

Now merge the vertex C of T_1 and the vertex C of T_2 to obtain the tree T.

Case 2: There is no pair of antipodal subgraphs w.r.t. C.

Subcase 2(c): Either there exist an incompatible pair (G_i, G_j), or there exist G_i, G_j, and G'_j s.t. $G_i > G_j$, $G_i > G'_j$, and $W(G_j) \ne W(G'_j)$.

Let $Y = X - \{ G_i\}$. Now Y satisfies the hypothesis of lemma 3.4.14. Let G_1^*, $G_2^*, \dots, G_t^*$ be an arrangement of Y according to lemma 3.4.14. Construct a tree T_1 for the collection Y using the technique as in case I. Let T''_i be a tree for G_i. Now merge the leaf vertex C of T''_i with the vertex C of T_1 to obtain T.

Subcase 2(d): Subcase 2(c) is not true.

Let G'_1, $G'_2,\ldots,G'_r$ be an ordering of the separated graphs of G according to the Lemma 3.4.14. Construct a tree T for this ordering using the technique as in case 1.

Next we prove that the tree T so obtained is a PV-clique tree for G. We consider case 1 and case 2 separately.

Assume that case 1 is true. To show that the tree T_1 is a PV-clique tree for $\bigcup_{i=1}^{t_1} G'_i$, it is enough to show that $T_1^{(k)}$ is a PV-clique tree for $\bigcup_{i=1}^{k} G'_i$, if $T_1^{(k-1)}$ is a PV-clique tree for $\bigcup_{i=1}^{k-1} G'_i$. If $|W(G'_k)| \geq 2$, then by the ordering of X, and by (ii) of Theorem 3.4.15, G'_k is not attached to any of the separated graphs G'_i, $1 \leq i \leq k-1$. If G'_k is not attached to any of the separated graphs G'_i, $1 \leq i \leq k-1$, then we merge the leaf vertex C of T'_k to the vertex C of $T_1^{(k-1)}$ to obtain $T_1^{(k)}$. So $I_{T_1^{(k)}}(C') \leq 1$ for all $C' \in C(\bigcup_{i=1}^{k} G'_i)$. Let $W(G'_k) = \{ v'_k\}$ and let G'_k be attached to some separated graph. Let i be the largest index such that G'_i is attached to G'_k, $1 \leq i \leq k-1$. Now either $G'_i > G'_k$ or $G'_i \sim G'_k$. So C', so chosen in subcase 1(b) of case 1, must belong to T'_i. Now according to the choice of T'_i, $I_{T'_i}(C') = 0$. Since Condition (v) (b) is true, $I_{T_1^{(k-1)}}(C') = 0$. Since T'_k and $T_1^{(k-1)}$ are PV-clique trees for G'_k and $\bigcup_{i=1}^{k-1} G'_i$, respectively $T_1^{(k)}$ is a PV-clique tree for $\bigcup_{i=1}^{k} G'_i$.

Similarly, T_2 is a PV-clique tree for $\bigcup_{i=1}^{t_2} G_i''$. Since $G_1 \Leftrightarrow G_2$, and $|W(G_1) \cap W(G_2)| = 1$, $I_T(C) \geq 1$. Again $I_{T_1}(C) = 0$ and $I_{T_2}(C)=0$. If possible, let $I_T(C) > 1$. Let C be an internal vertex to a path $\pi(w)$, $w \neq v$. Then there exist separated graphs G'_i and G''_j s.t. $w \in W(G'_i) \cap W(G''_j)$. Since $G_2 \geq G''_j$, $W(G'_j) \cap W(G_2) \neq \phi$. So by the construction of Y, $G'_i \Leftrightarrow G_2$.

So $G_1'=G_1$. So $w = v$, a contradiction. Hence $I_T(C) = 1$. Thus T is a PV-clique tree for G.

We next consider case 2. Assume that Subcase(c) is true. Since Proposition 3.4.13 (3) and (4) are true, and T_1 is constructed as in the construction of the T_1 in case 1, T_1 is a PV-clique tree for $\bigcup_{i=1}^{t} G_i^*$. Now in both T_1 and T_i^*, C is a leaf vertex. since $|W(G_j)| =1$, C will be an internal vertex of the path $\pi(v_j)$ in T, where $\{v_j\}=W(G_j)$. So $I_T(C) = 1$. Hence T is a PV-clique tree for G.

In subcase 2(d), the tree T is constructed following the technique for the construction of the T_1 in case 1. So the tree T constructed in this subcase is a PV-clique tree for G. So G is a PV-graph.∎

3.5 Forbidden Subgraph Characterization of PV graphs:

In this section we provide the forbidden subgraph characterization for PV-graphs. To this end we need some lemmas.

Lemma 3.5.1: Let G_1 be a separated graph of a chordal graph G w.r.t. C s.t. G_1 is a PV-graph. Then G_1 is incompatible w.r.t. v, $v \in W(G_1)$ iff at least one of the following conditions holds.

(1) There exists a pair (C_1', C_1'') of antipodal cliques w.r.t. a relevant clique C_1 of G_1 containing v, relevant to a vertex $v_1 \neq v$. (2) There exists a separated graph G_1' w.r.t. a relevant clique C_1 of G_1 containing v s.t. $N(D(G_1')) = 2$.

(3) There exists an incompatible pair (G_1', G_1'') of separated graphs w.r.t. a relevant clique C_1 of G_1 containing v s.t. $v_1' \neq v$.

Proof: Necessity:

Let T_1 be a PV-clique tree for G_1, and $C_1 \neq C$ be an end vertex of the path $\pi(v)$ in T_1. Since G_1 is incompatible w.r.t v, $I_{T_1}(C_1)=1$. So C_1 is a separating clique of G_1. If possible, let none of the conditions (1), (2), and (3) of Lemma 3.5.1 hold for C_1. Since the intersection of any three

cliques of G_1 is a singleton set, there does not exist a pair (G_1', G_1'') of antipodal separated graphs w.r.t. C_1, because condition (1) does not hold. So the collection of all separated graphs w.r.t. C_1 satisfies the condition of Lemma 3.4.14. Now as in Case 1 of the sufficiency of Theorem 3.4.15, we can construct a PV-clique tree T_1^* s.t. $I_{T_1^*}(C_1) = 0$. This is a contradiction as G_1 is incompatible w.r.t. v. So at least one of the conditions (1), (2), and (3) holds.

Sufficiency:

Let T_1 be any PV-clique tree for G_1 s.t. $C_1 \neq C$ is an end vertex of $\pi(v)$ in T_1. If (1) holds, then by Proposition 3.4.8, C_1' and C_2' belong to two different separated graphs. So if (1) holds, then by Proposition 3.4.6 $I_{T_1}(C_1)=1$. If (2) holds, then as in the proof of Lemma 3.4.13(2), $I_{T_1}(C_1)=1$. If (3) holds, then by Proposition 3.4.12, $I_{T_1}(C_1)= 1$. So G_1 is incompatible w.r.t. v.∎

Let G_i be incompatible w.r.t. v, $v \in W(G_i)$. Let T_i be a PV-clique tree for G_i and $C_i \neq C$ be an end vertex of the path $\pi(v)$ in T_i. So clearly $I_{T_i}(C_i) = 1$. Define depth(G_i,v) = 1 if either (i) there exists a pair of antipodal cliques w.r.t. C_i relevant to a vertex v_i s.t. $v_i \neq v$, or (ii) for the separated graph G_1' w.r.t. C_i containing C, $N(D(G_1'))=1$, or there exists a separated graph G_i' w.r.t. C_i s.t. $N(D(G_i'))=2$. Suppose depth(G_i,v) >1. Then by Lemma 3.5.1 there exists an incompatible pair (G_i', G_j') w.r.t. C_i. Let $W(G_i') \cap W(G_j') = \{v_j'\}$. If $G_i' > G_j'$, then depth(G_i,v)= 1+depth(G_i', v_j); otherwise, depth(G_i,v)= 1 + Max.{ depth(G_i' ,v_j), depth(G_j',v_j) }.

Let (G_i,G_j) be an incompatible pair w.r.t. C. Then height(G_i,G_j)= depth(G_i,v_j) if $G_i>G_j$; otherwise, height(G_i,G_j)= max{ depth(G_i,v_j), depth(G_j,v_j)}, where $\{v_j\}= W(G_i) \cap W(G_j)$.

Let $\mathcal{F}_{\mathcal{T}}$ = { H s.t. H is a minimal forbidden subgraph for PV-graph }.

Let $\mathcal{F}_{\mathcal{T}_1}$ = {$H \in \mathcal{F}_{\mathcal{T}}$, and H has a separating clique C s.t. if (G_1, G_2) is any incompatible pair w.r.t. C, then height$(G_1,G_2) = 1$ }.

Lemma 3.5.2: The intersection of any three cliques in a chordal graph G is at most a singleton set iff G does not contain H_1 and H_2 in Figure 3.5.1 as induced subgraphs.

Proof: Necessity:

If possible, let G contain H_1(H_2) as an induced subgraph. Let C_1, C_2, and C_3 be any ordering of the cliques of $H_1(H_2)$. Let C_1', C_2', and C_3' be some cliques of G containing C_1, C_2, and C_3 of H_1(H_2), respectively. Now $|C_1' \cap C_2' \cap C_3'| \geq 2$, as $|C_1 \cap C_2 \cap C_3| \geq 2$, contrary to our assumption.

Sufficiency:

If possible, let C_1, C_2, and C_3 be in G s.t. $|C_1 \cap C_2 \cap C_3| \geq 2$. Let x, y $\in C_1 \cap C_2 \cap C_3$.

Case 1: There exists an ordering, say (C_{i_1}, C_{i_2} , C_{i_3}) s.t. $(C_{i_1} \cap C_{i_2}) \cup (C_{i_3} \cap C_{i_2}) = C_{i_2}$.

By the maximality of C_{i_2} there exist $x_1 \in C_{i_2} - C_{i_1}$ and $x_1' \in C_i-(C_i \cup C_{i_3})$ s.t. $x_1x_1' \notin E(G)$. Similarly there exist $x_2 \in C_{i_2} - C_{i_3}$ and $x_2' \in C_{i_3}-(C_{i_1} \cup C_{i_2})$ s.t. $x_2x_2' \notin E(G)$. Due to case 1, $x_1 \neq x_2$, $x_1'x_2 \in E(G)$, and $x_2'x_1 \in E(G)$. Now $x_1'x_2' \notin E(G)$, otherwise x_1', x_2, x_1, x_2' will form a chordless 4-cycle in the chordal graph G. Now $G' = G[\{x, y, x_1, x_2, x_1', x_2'\}]$ is isomorphic to H_2.

Case 2: There is no ordering $(C_{i_1}, C_{i_2}, C_{i_3})$ of C_1, C_2, and C_3 s.t. $(C_{i_1} \cap C_{i_2}) \cup (C_{i_2} \cap C_{i_3}) = C_{i_2}$.

If possible, let $(C_1 \cup C_2) - C_3$ induce a complete subgraph of G. Now $(C_1 \cap C_3) \neq (C_2 \cap C_3)$, otherwise $(C_1 \cup C_2)$ will induce a complete subgraph of G, a contradiction to the maximality of C_1. Let $x_1 \in (C_1 \cap C_3)-(C_2 \cap C_3)$

and $x_2 \in (C_2 \cap C_3) - (C_1 \cap C_3)$. So there exist $y_1 \in C_2-C_3$ and $y_2 \in C_1-C_3$ s.t. $x_1y_1 \notin E(G)$ and $x_2y_2 \notin E(G)$. Since $(C_1 \cup C_2)-C_3$ induces a complete subgraph, $G[\{x_1,y_1,x_2,y_2\}]$ is a chordless 4-cycle of G, a contradiction. So $(C_1 \cup C_2)-C_3$ does not induce a complete subgraph. Let $v_1 \in C_1-C_2$ and $v_2 \in C_2-C_1$ such that $v_1v_2 \notin E(G)$. If there exists $z \in C_3$ s.t. $z\,v_1 \notin E(G)$ and $zv_2 \notin E(G)$, then $G[\{x,y,v_1,v_2,z\}]$ is isomorphic to H_1. Assume that for every $z \in C_3$ either $zv_1 \in E(G)$ or $zv_2 \in E(G)$. Due to the maximality of C_3 there exist z_1 and z_2 in C_3 s.t. $z_1v_1 \notin E(G)$ and $z_2v_2 \notin E(G)$. Then $z_1v_2 \in E(G)$, $z_2v_1 \in E(G)$, and $G[\{x,y,v_1,v_2,z_1,z_2\}]$ is isomorphic to H_2.

Hence the intersection of any three cliques in G is at most a singleton set.▮

Lemma 3.5.3: Let G be a chordal graph free from $H_1$and H_2. Then an induced subgraph of G has two pairs of antipodal subgraphs w.r.t. some separating clique C, iff G contains one of H_3, H_4, H_5 and H_6 in Figure 3.5.1 as an induced subgraph.

Proof: Sufficiency:

It is easy to verify that each of the graphs in Figure 3.5.1 has two pairs of antipodal subgraphs w.r.t. a separating clique.

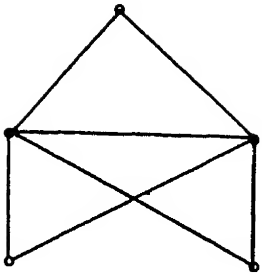
Necessity:

Wlg, G has two pairs of antipodal subgraphs w.r.t a separating clique C.

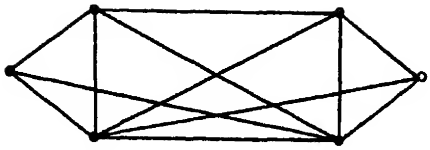
Case 1: There exist pair wise antipodal subgraphs G_1, G_2, G_3.

Now, by Proposition 3.4.5, there exists C_i in G_i, i=1, 2, 3 s.t. $C_i \Leftrightarrow C_j$ iff $1 \le i \ne j \le 3$. since G is free from H_1 and H_2, $|C_1 \cap C_2 \cap C_3| \le 1$.

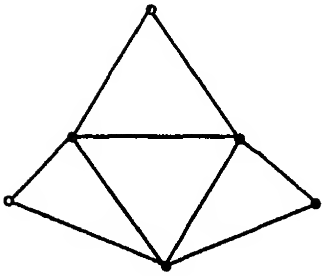
When $|C_1 \cap C_2 \cap C_3| = 1$, let $C_1 \cap C_2 \cap C_3 = \{x\}$, and let $\{x,y_i\} \subset C_i \cap C$ be s.t. $y_i \notin C_j$, with $i \ne j$ and $1 \le i,j \le 3$. Let x_i be in $C_i \setminus C$, $1 \le i \le 3$. Then $G[\{x, x_1, x_2, x_3, y, y_1, y_2, y_3\}]$ is isomorphic to H_4. If $|C_1 \cap C_2 \cap C_3| = 0$, let x, y, z, x_1, x_2, and x_3 be in $C_1 \cap C_2$, $C_2 \cap C_3$,



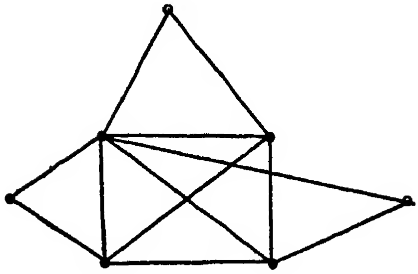
H_1



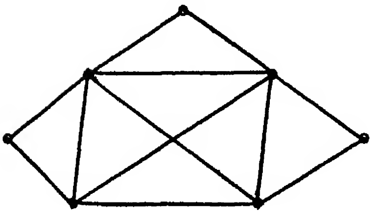
H_2



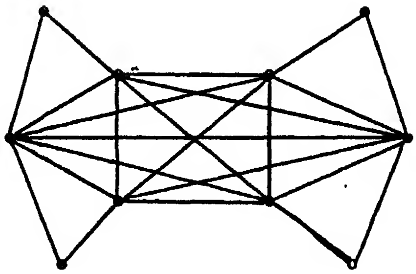
H_3



H_4



H_5



H_6

Figure 3.5.1: Some Forbidden subgraphs for PV-graphs.

$C_3 \cap C_1$, $C_1 \setminus C$, $C_2 \setminus C$ and $C_3 \setminus C$, respectively. Then $G[\{x, y, z, x_1, x_2, x_3\}]$ is isomorphic to H_3.

Case 2: There exist G_1, G_2, and G_3 satisfying $G_1 \Leftrightarrow G_2$, $G_1 \Leftrightarrow G_3$, but G_2 is not antipodal to G_3.

By Proposition 3.4.5, there exists C_i in G_i, i=1, 2,3 s.t. $C_1 \Leftrightarrow C_2$, $C_1 \Leftrightarrow C_3$ but C_2 is not antipodal to C_3. As G is free from H_1 and H_2, we have $C_2 \cap C_3 = \phi$, $|C_1 \cap C_2| = 1$ and $|C_1 \cap C_3| = 1$. Taking $v \in C_1 \cap C_2$, $w \in C_1 \cap C_3$, $\{v,v'\} \subset C \cap C_2$, $\{w,w'\} \subset C \cap C_3$, x_i in $C_i \setminus C$, $1 \le i \le 3$, we get $G[\{x_1,x_2,x_3,v, v', w, w'\}]$ which is isomorphic to H_5.

Case 3: There exist G_1, G_2, G_3, and G_4 all distinct s.t. $G_1 \Leftrightarrow G_2$ and $G_3 \Leftrightarrow G_4$.

By Proposition 3.4.5, there exists C_i in G_i, i=1,2,3,4 s.t. $C_1 \Leftrightarrow C_2$ and $C_3 \Leftrightarrow C_4$. Now G being free from H_1 and H_2, we have $|C_1 \cap C_2| = 1$ and $|C_3 \cap C_4| = 1$. Let $\{x\} = C_1 \cap C_2$, $\{y\} = C_3 \cap C_4$, $\{y, y'\} \subset C_3 \cap C$, $\{x, x'\} \subset C_1 \cap C$, $\{x, x''\} \subset C_2 \cap C$, $\{y, y''\} \subset C_4 \cap C$, and $z_i \in C_i \setminus C$, i= 1, 2, 3, 4. Then $G[\{x, x', x'', y, y', y'', z_1, z_2, z_3, z_4\}]$ is isomorphic to H_6. ∎

Lemma 3.5.4: Let G_1 be a separated graph of G w.r.t. C s.t. G_1 is a PV-graph, G_1 is incompatible w.r.t. v, $v \in W(G_1)$, and $\mathrm{depth}(G_1,v) = 1$. Then the following holds.

(a) If $|W(G_1)| = 1$, then $G_1-(C-W(G_1))$ contains a subgraph isomorphic to one of the graphs $H'_1,\dots,H'_4$ of Figure 3.5.2.

(b) If $|W(G_1)| \ge 2$, then $G_1-(C-W(G_1))$ contains a subgraph isomorphic to one of the graphs H'_5 to H'_9 in Figure 3.5.2.

Proof:

Let T_1 be a PV-clique tree for G_1, and $\pi(v)= C,C_1,\dots,C_i$. Let G'_1 be the separated graph of G_1 w.r.t. C_i containing C. Let $v_j \in C_j \cap C_{j+1}$ s.t. $v_j \ne v$, $1 \le j \le i-1$. Now $\mathrm{depth}(G_1,v)=1$. So at least one of the following

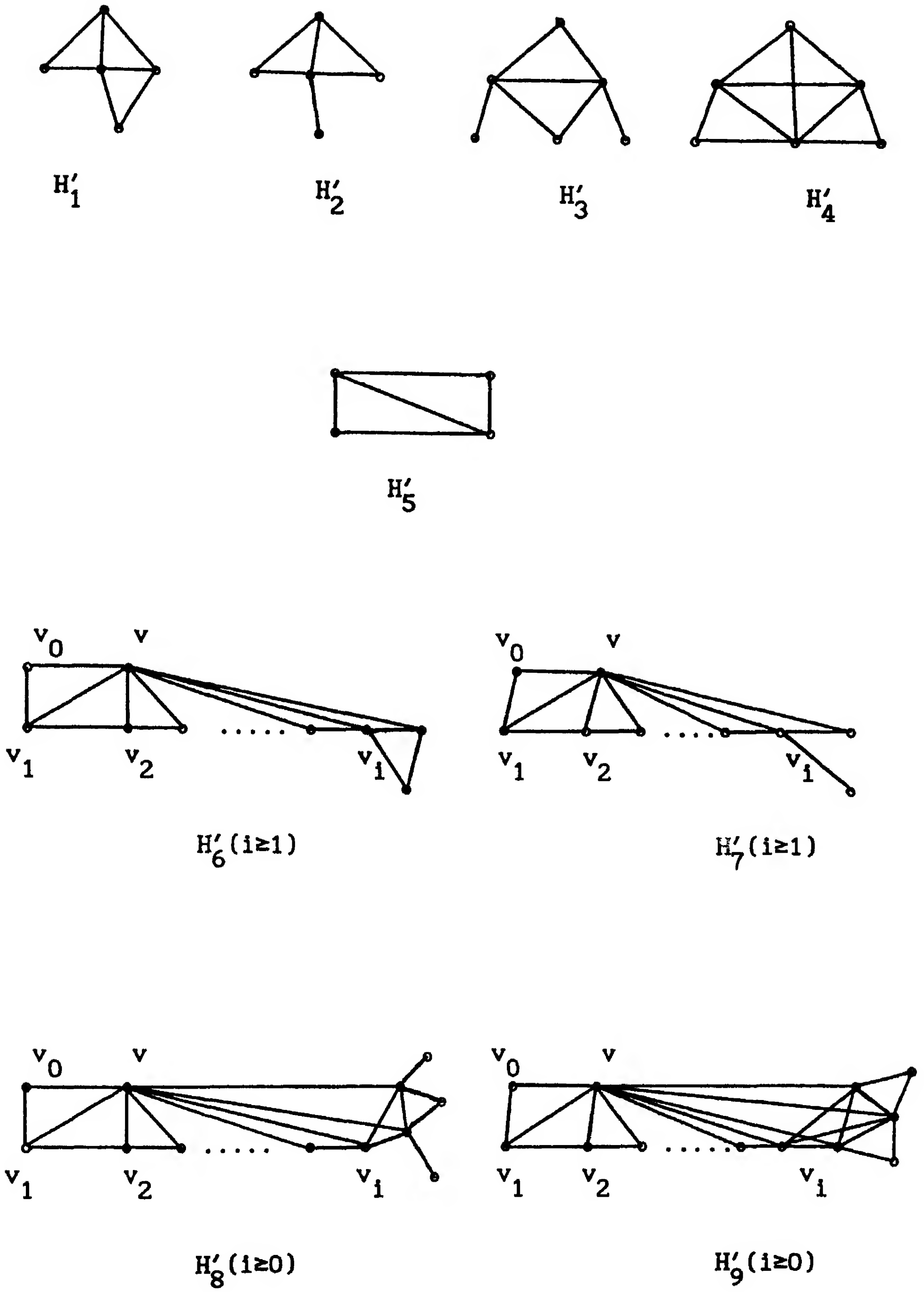


Figure 3.5.2: Incompatibility in terms of forbidden subgraphs.

three cases holds.

Case 1: There exists a pair (C_i', C_i'') of antipodal cliques w.r.t.C_i.

Assume that $|W(G_1)| \geq 2$. Let $w \in W(G_1)-v$.

If there exists a relevant clique of G_1 other than C_1 that intersects $W(G_1)-v$, then i=1, as T_1 is a PV-clique tree for G_1. Wlg, Let w lie in a relevant clique of G_1 other than C_1. Let $\pi(w)= C,C_1,C_2', \ldots C_k'$. Now clearly C_1 is a separating clique of G_1, and (C, C_2') is a pair of antipodal cliques w.r.t. C_1 relevant to v_1. Let $w_1 \in C_2' -C_1$, and $w_2 \in (C_1 \cap C_2')-W(G_1)$. Then $G[\{v,w,w_1, w_2,\}]$ is isomorphic to H_5'. So assume that there is no relevant cliques of G_1 other than C_1 that intersects $W(G_1)-v$.

Now because of Proposition 3.4.8, C_i' and C_i'' lie in different separated graphs of G_1 w.r.t. C_i. If G_1' contains one of C_i' and C_i'', say C_i' , then C_i' must be the principal clique of G_1'. So $C_i'= C_{i-1}$. Now choose v_{i-1} s.t. $v_{i-1} \in (C_{i-1} \cap C_i \cap C_i'')$. Let $y_1 \in ((C_i \cap C_i'') - W(G_1'))$, and $y_2 \in C_i''- C_i$. Then $G[\{v,w,v_1,v_2,\ldots, v_{i-1},y_1,y_2\}]$ is isomorphic to H_6'. If G_1' contains neither C_i' nor C_i'', then clearly, C_i' and C_i'' are unattached to C_{i-1}; otherwise, (C_{i-1}, C_i'), where wlg, C_i' is attached to C_{i-1}, is an antipodal pair w.r.t. C_i. let $y_1 \in (C_i \cap C_i')- C_i''$, $y_2 \in (C_i \cap C_i'')-C_i'$, $y_3 \in (C_i \cap C_i' \cap C_i'')$, $w_1 \in C_i'-C_i$, $w_2 \in C_i''-C_i$. Then $G[\{v,w,w_1,w_2,v_1,v_2,\ldots,v_{i-1},y_1,y_2,y_3\}]$ is isomorphic to H_7'.

Now assume that $|W(G_1)|=1$.

As above C_i' and C_i'' lie in different separated graphs of G_1 w.r.t. C_i. If G_1' contains one of C_i' and C_i'', say C_i', then C_i' must be the principal clique of G_1'. So $C_i'=C_{i-1}$. Let $y \in C_{i-1}-C_i$, $y_1 \in C_{i-1} \cap C_i \cap C_i''$, $y_2 \in ((C_i \cap C_i'') - W(G_1'))$, and $y_3 \in C_i''- (C_i'' \cap C_i)$. Then $G[\{v,y,y_1,y_2,y_3\}]$ is isomorphic to H_1'. If G_1' contains neither C_i' nor C_i'', then let $y_1 \in (C_i \cap C_i')- C_i''$, $y_2 \in (C_i \cap C_i'')-C_i'$, $y_3 \in (C_i \cap C_i' \cap C_i'')$, $y_4 \in C_i'-C_i$, $y_5 \in C_i''-C_i$. Then $G[\{v, y_1,y_2,y_3, y_4,y_5\}]$ is isomorphic to H_2'.

Case 2: There exists a separated graph G'_i w.r.t. C_i s.t. $N(D(G'_i))=2$.

Let $G'_i > G'_j$, and $G'_i > G_j''$ be s.t. $v'_j \neq v_j''$, where $\{ v'_j\} = W(G'_j)$, and $\{v_j''\}= W(G_j'')$. Let $y_1 \in C'_i-C_i$, $y_2 \in C'_j-C_i$, and $y_3 \in C_j''-C_i$, where C'_i, C'_j, and C_j'' are principal cliques of G'_i, G'_j, and G_j'', respectively. If $|W(G_1)|=1$, then $G[\{v,y_1,y_2,y_3, v'_j,v_j''\}]$ is isomorphic to H'_3. If $|W(G_1)|\geq 2$, then let $w \in W(G_1)-v$. Then $G[\{v,w,y_1,y_2,y_3,v'_j,v_j'', v_1,\dots,v_{i-1}\}]$ is isomorphic to H'_8.

Case 3: $N(D(G'_i))= 1$.

We can choose v_{i-1} s.t. $\{v_{i-1}\}= W(G'_i)$, where $G'_1> G'_i$, and G'_i is a separated graph of G_1 w.r.t. C_i. Let $y_1 \in C_i- W(G'_i)$, and $y_2 \in C'_i-C_i$, where C'_i is a principal clique of G'_i. If $|W(G_1)| \geq 2$, then let $w \in W(G_1)-v$. Then $G[\{ v,w,v_1,v_2,\dots, v_{i-1},y_1, y_2\}]$ is isomorphic to H'_9. If $| W(G_1) | = 1$, then let $y_3 \in C_{i-1}- C_i$. Then $G[\{ v, y_1,y_2,y_3,v_{i-1}\}]$ is isomorphic to H'_4.∎

Let H'_i, $1\leq i\leq 9$ be the graph in Figure 3.5.2.

Let $f_1(H'_i,H'_j,H'_k,H'_s)$, where $1\leq i,j,k,s \leq 4$, and i,j,k, and s need not be distinct, be the graph H which is obtained as follows:

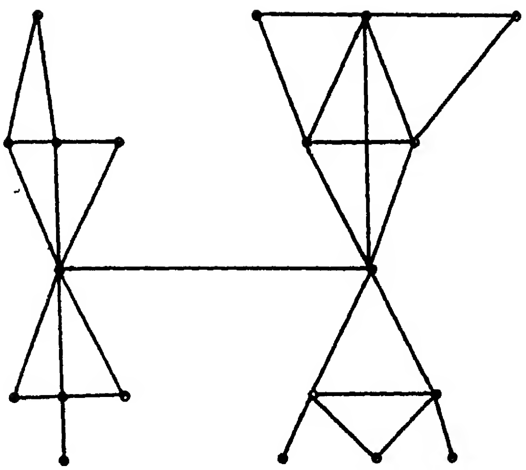
(1) Merge the vertex v of H'_i and the vertex v of H'_j, call the new vertex v' and the new graph H'.

(2) Merge the vertex v of H'_k and the vertex v of H'_s, call the new vertex v'' and the new graph H''.

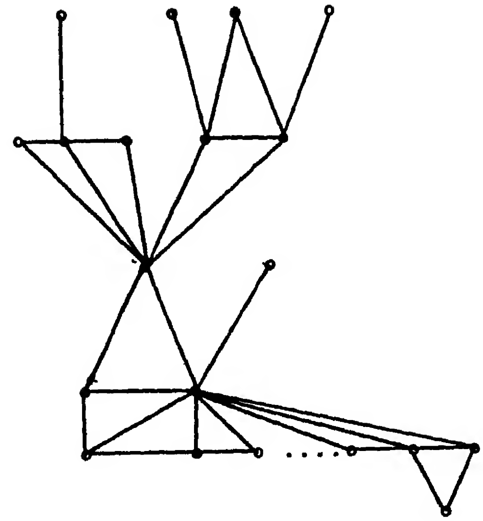
(3) Join the vertex v' of H' to the vertex v'' of H'' by an edge and the resulting graph is called H.

Let $f_2(H'_i,H'_j,H'_k)$, where $5\leq i\leq 9$, and $1\leq j,k \leq 4$, and j,k need not be distinct, be the graph H which is obtained as follows:

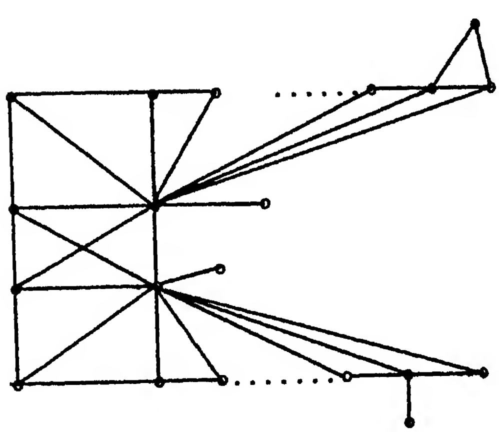
(1) Merge the vertex v of H'_j and the vertex v of H'_k, call the new vertex v' and the new graph H'. (2) Take a new vertex v'' and join it to the vertex v of G'_i by an edge, and call the new graph H''. (3) Join the vertices v and v_0 of H'' to the vertices v' of H' by two edges. The new graph is called H.



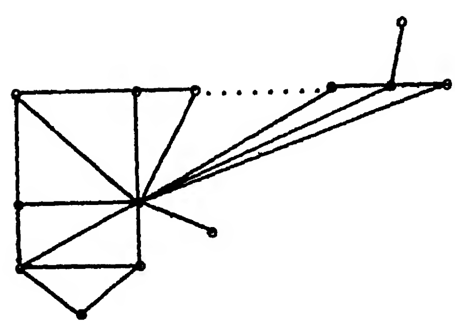
$f_1(H'_1, H'_2, H'_3, H'_4)$



$f_2(H'_6, H'_2, H'_3)$



$f_3(H'_6, H'_7)$



$f_4(H'_5, H'_7)$

Figure 3.5.3: An Illustration of the operations of the functions f_1 to f_4.

Let $f_3(G'_i, G'_j)$, $5 \le i, j \le 9$, and i, j need not be distinct, be the graph obtained as follows:

(1) Take a new vertex v" and join it to the vertex v of H'_i, call the vertices v and v_0 as v′ and w'_0, respectively, and call the new graph H′. (2) Take a new vertex v‴ and join it to the vertex v of H'_j, call the new graph H". (3) Join the vertices v′ and v'_0 of H′ with the vertices v and v_0 of H". The new graph is called H.

Let $f_4(H'_i, H'_j)$, $5 \le i, j \le 9$, where i and j need not be distinct, be the graph obtained as follows:

(1) Call the vertex v and v_0 of H'_j as v′ and v'_0. Merge the vertex v of H'_i and the vertex v′ of H'_j, call the new vertex v′, and the new graph H′. Join the vertex v_0 and v'_0 of H′. Take a new vertex v" and join it to the vertex v of H′, and call the new graph H.

Let $S_1 = \{ H \text{ s.t. } H=f_1(H'_i,H'_j,H'_k,H'_s),\ 1 \le i,j,k,s \le 4\}$.
Let $S_2 = \{ H \text{ s.t. } H=f_2(H'_i,H'_j,H'_k),\ 5\le i\le 9,\ 1\le j,k \le 4 \}$. Let $S_3=\{ H \text{ s.t. } H=f_3(G'_i,G'_j),\ 5 \le i < j \le 9 \}$. Let $S_4= \{H \text{ s.t. } H=f_4(H'_i, H'_j),\ 5 \le i,j \le 9\}$.

The operations of the functions f_1 to f_4 are illustrated in Figure 3.5.3.

Theorem 3.5.5: A graph $G \in \mathcal{F}_{\mathcal{T}_1}$ iff G is isomorphic to either one of H_1 to H_6 of Figure 3.5.1 or one of the graphs H_7 to H_{39} in Figure 3.5.4 or a member of S_1, S_2, S_3, or S_4.

Proof: It is a routine exercise to check using Theorem 3.4.15 that each of the graphs mentioned in Theorem 3.5.5 belongs to $\mathcal{F}_{\mathcal{T}_1}$.

Necessity :

Let G be in $\mathcal{F}_{\mathcal{T}_1}$. If G is not chordal, then G contains C_n, $n \ge 4$, i.e. H_7 as an induced subgraph. Since $H_7 \in \mathcal{F}_{\mathcal{T}_1}$, G will be isomorphic to H_7. Assume that G is a chordal graph. Clearly G has a separating clique. Let C

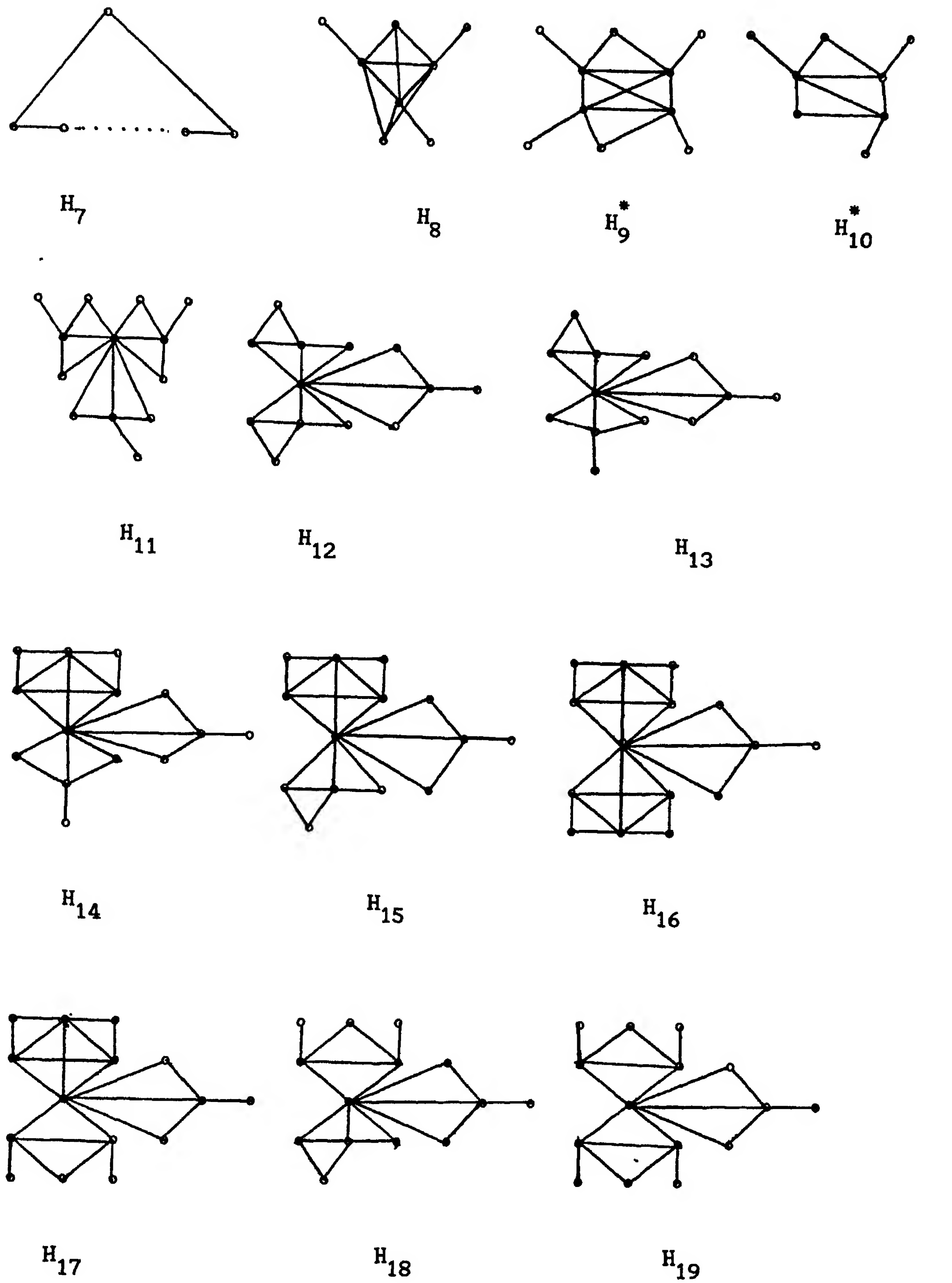
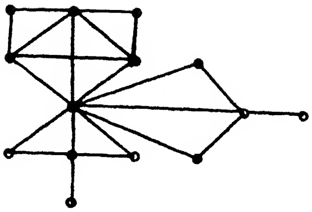
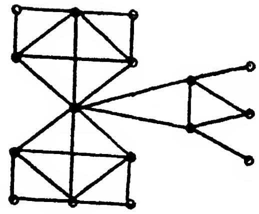


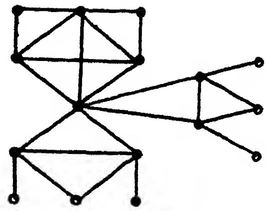
Figure 3.5.4: Forbidden Subgraphs for PV-graphs.



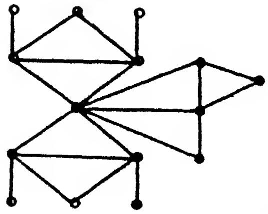
H_{20}



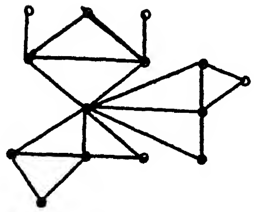
H_{21}



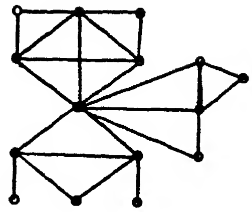
H_{22}



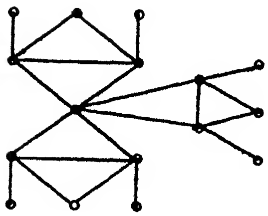
H_{23}



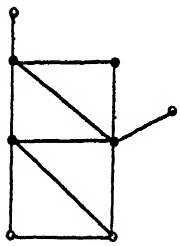
H_{24}



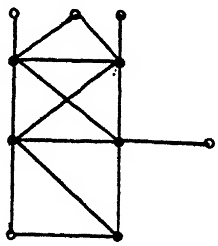
H_{25}



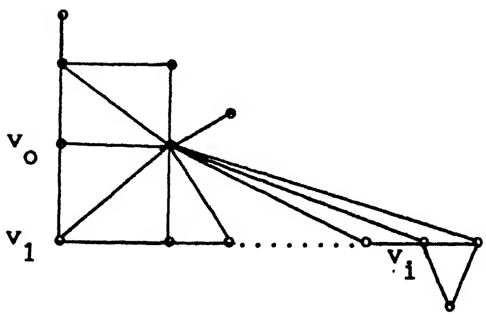
H_{26}



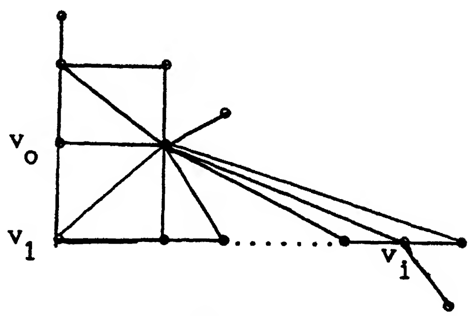
H^{*}_{27}



H^{*}_{29}

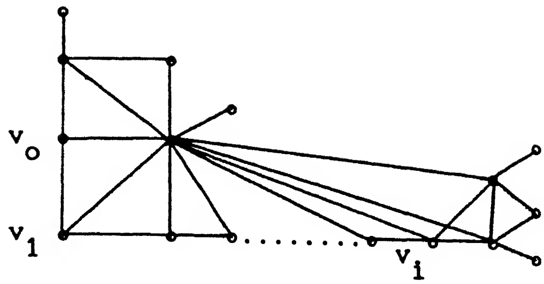


$H^{**}_{27}(i \geq 1)$

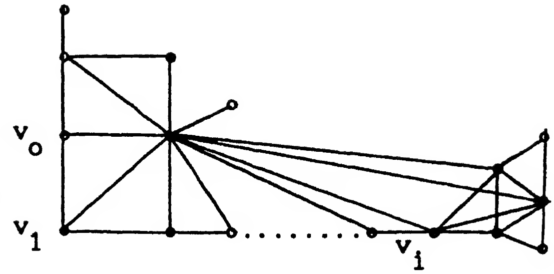


$H^{**}_{10}(i \geq 1)$

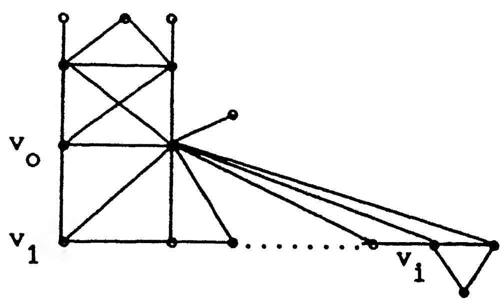
(Figure 3.5.4 Continued)



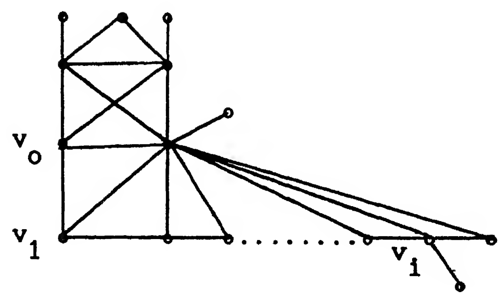
H_9^{**} ($i \geq 0$)



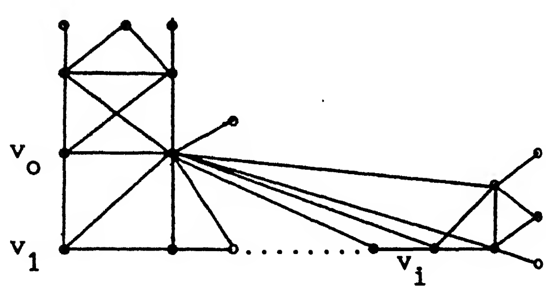
H_{28}^{*} ($i \geq 0$)



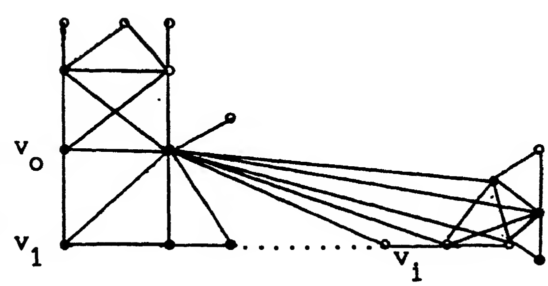
H_{29}^{**} ($i \geq 1$)



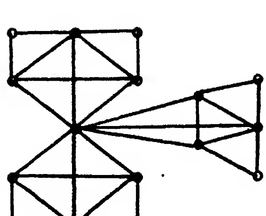
H_9^{***} ($i \geq 1$)



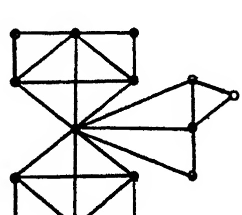
H_{30} ($i \geq 0$)



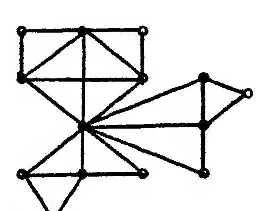
H_{31} ($i \geq 0$)



H_{32}

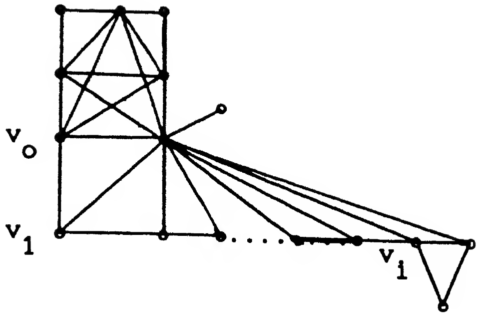


H_{33}

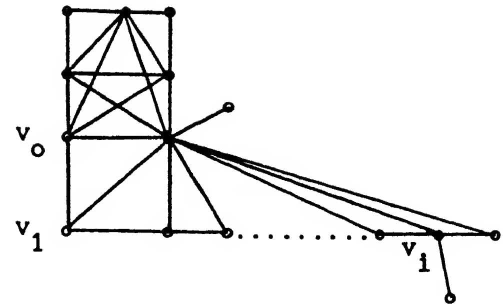


H_{34}

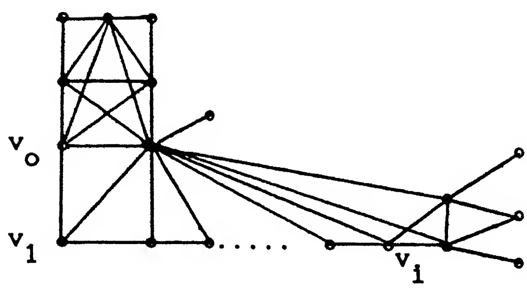
(Figure 3.5.4 continued)



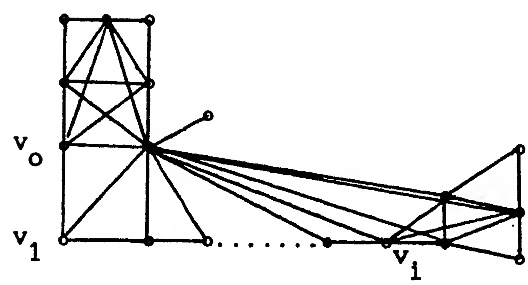
$H_{35}^{**}(i\geq1)$



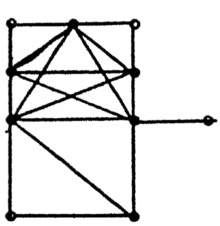
$H_{28}^{**}(i\geq1)$



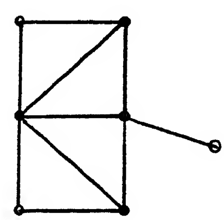
$H_{31}^{*}(i\geq0)$



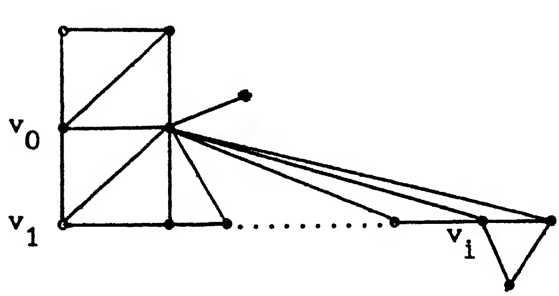
$H_{36}(i\geq0)$



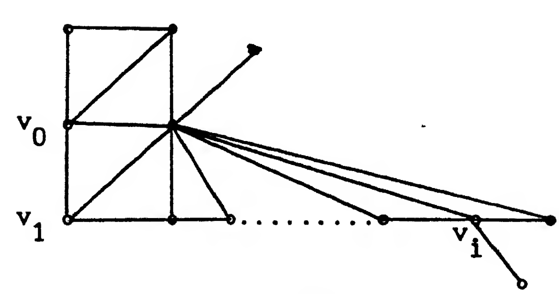
H_{35}^{*}



H_{37}

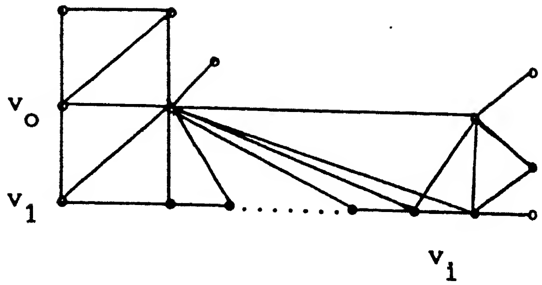


$H_{38}(i\geq1)$

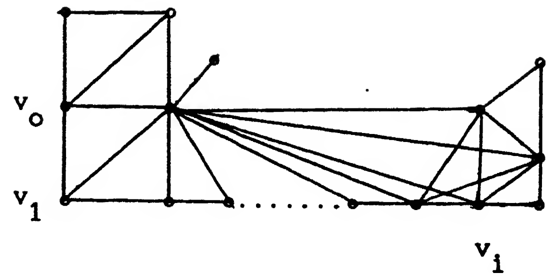


$H_{27}^{***}(i\geq1)$

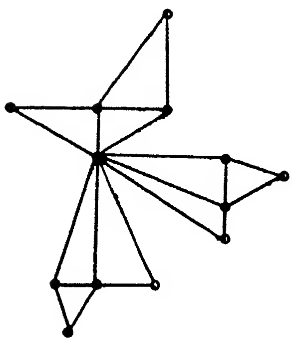
(Figure 3.5.4 continued)



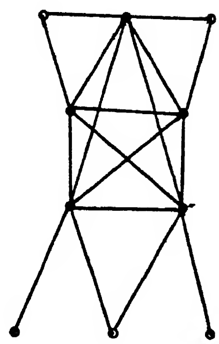
$H_{29}(i \geq 0)$



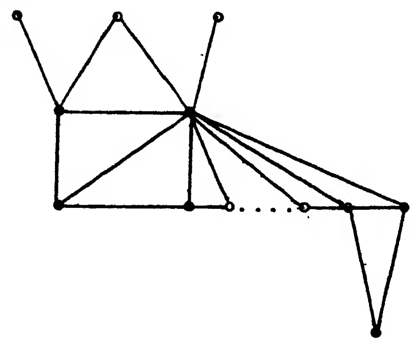
$H_{35}(i \geq 0)$



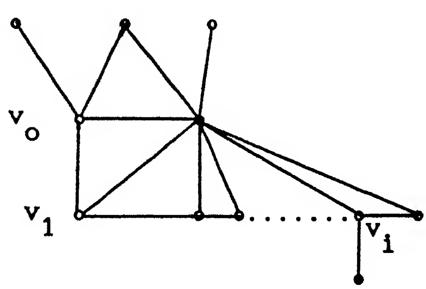
H_{39}



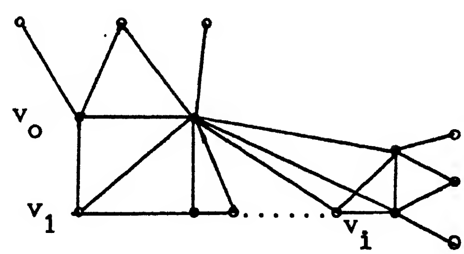
H_{28}^{***}



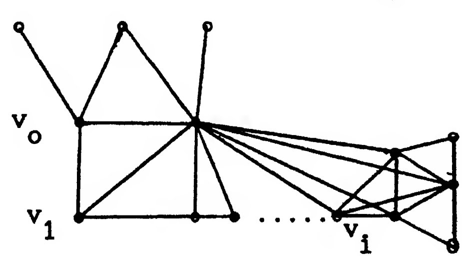
$H_{27}(i \geq 1)$



$H_{10}(i \geq 1)$



$H_9(i \geq 0)$



$H_{28}(i \geq 0)$

Figure 3.5.4: Forbidden Subgraphs for PV-Graphs.

be a separating clique of G satisfying the property in the definition of $\mathcal{F}_{\mathcal{J}_1}$. Let G_i, $1 \le i \le r$, $r \ge 2$ be the separated graphs of G w.r.t. C. Since G is not a PV-graph, Theorem 3.4.15 will not hold for G. Clearly each G_i is a PV-graph.

Case 1: Theorem 3.4.15 (2) is not true.

Then by Lemma 3.5.2, G will contain a subgraph isomorphic to H_1 or to H_2. since both H_1 and H_2 are in F_{S_1}, G itself will be isomorphic to H_1 or to H_2.

Case 2: Theorem 3.4.15 (3) is not true.

Then by Lemma 3.5.3 and by the fact that $H_i \in \mathcal{F}_{\mathcal{J}_1}$, $3 \le i \le 6$, G will be isomorphic to one of H_3 to H_6.

Case 3:: Theorem 3.4.15 (4) is not true.

Subcase 3(a): There exists a G_i s.t. $N(D(G_i)) \ge 3$.

Let G_j, G'_j, and G_j'' belong to $D(G_i)$ s.t. v_j, v'_j, and v_j'' are all distinct, where $\{v_j\} = W(G_j)$, $\{v'_j\} = W(G'_j)$, and $\{v''_j\} = W(G_j'')$. Let $x_1 \in C_i$, $x_2 \in C_j$, $x_3 \in C'_j$, and $x_4 \in C''_j$, where C_i, C_j, C'_j, and C''_j are some principal cliques of G_i, G_j, G'_j, and G_j'', respectively. Let $x \in C-W(G_i)$. Then $G[\{x, x_1, x_2, x_3, x_4, v_j, v'_j, v_j''\}]$ is isomorphic to H_8. Since $H_8 \in \mathcal{F}_{\mathcal{J}_1}$, G is isomorphic to H_8.

Subcase 3(b): There exist G_1 and G_2 s.t. $N(D(G_i)) = 2$, for i=1,2.

Let $G_3, G_4 \in D(G_1)$ and $G_5, G_6 \in D(G_2)$ s.t. $v_3 \ne v_4$ and $v_5 \ne v_6$, where $\{v_i\} = W(G_i)$, $1 \le i \le 6$. If $G_1 | G_2$, then let $x_i \in C-C_i$, where C_i is a principal clique of G_i, $1 \le i \le 6$. Then $G[\{v_3, v_4, v_5, v_6, x_1, x_2, x_3, x_4, x_5, x_6\}]$ is isomorphic to H_9^*. Since $H_9^* \in \mathcal{F}_{\mathcal{J}_1}$, G is isomorphic to H_9^*. Assume that G_1 is attached to G_2. Since $|W(G_i)| \ge 2$, for i=1, 2, $G_1 \Leftrightarrow G_2$, otherwise this case reduces to Case 1. If v_3, v_4, v_5, and v_6 are all distinct, then $G'=G-v$, where $\{v\} = W(G_1) \cap W(G_2)$, is not a PV-graph, since C-v is a separating

clique of G' and Proposition 3.4.13 (2) is not true for G'. So, wlg, $v_3=v_5$. Then $G[\{v_3,v_4,v_6,x_1,x_2,x_3,x_4,x_6\}]$ is isomorphic to H^*_{10}. Since $H^*_{10} \in \mathcal{F}_{\mathcal{G}_1}$, G will be isomorphic to H^*_{10}.

Subcase 3(c): Proposition 3.4.13(3) is violated.

Let $G_1>G_2$, and (G_3,G_4) be an incompatible pair s.t. $G_1>G_3$, $G_1>G_4$ and $W(G_2)\neq W(G_3)$. Let $\{v\}= W(G_3)=W(G_4)$, $\{v_1\}=W(G_2)$, and $x_i \in C_i-C$, where C_i is a principal clique of G_i, i=1,2. Now each of $G_3-(C-v)$ and $G_4-(C-v)$ contains one of the graphs H'_1 to H'_4 as an induced subgraph. So $G[\,\{x_1,x_2,v_1\} \cup (V(G_3) \cup V(G_4)-C)\}]$ will contain one of the graphs H_{11} to H_{20} as an induced subgraph. Since $H_i \in \mathcal{F}_{\mathcal{G}_1}$, $11 \leq i \leq 20$, G is isomorphic to one of H_{11} to H_{20}.

Subcase 3(d): Proposition 3.4.13 (4) is not true.

Let $G_1>G_2$, $G_1>G_3$, $W(G_2)=\{x_2\}$, $W(G_3)=\{x_3\}$ s.t. $x_2 \neq x_3$. Let (G_4,G_5) be an incompatible pair. Suppose $G_4\sim G_5$. Then G_1 does not dominate G_4; otherwise, this case reduces to subcase 3(c). Assume that $G_1|\, G_4$. Now using a similar argument as in subcase 3(c), it can be seen that G is isomorphic to one of H_{17} to H_{26}.

Let $G_4>G_5$. If G_1 is attached to G_4, then $G_1 \Leftrightarrow G_4$. Again, either $x_2\in W(G_4)$ or $x_3\in W(G_4)$; otherwise, G-v, where $v \in W(G_1) \cap W(G_4)$ will not be a PV-graph.. Hence either $G_2 = G_5$ or $G_3= G_5$. Since $G-(C-W(G_4))$ is isomorphic to one of G'_5 to G'_9, G will be isomorphic to one of H^*_{27}, H^{**}_{27}, H^*_9, H^*_{10}, and H^*_{28}. If $G_1|\, G_4$, then G will be isomorphic to one of H^*_{29}, H^{**}_{29}, H^*_9, H_{30}, and H_{31}, since $G-(C-W(G_4))$ is isomorphic to one of H'_5 to H'_9 as an induced subgraph.

Case 4: Theorem 3.4.15(v) is not true.

Let (G_1,G_2) be an antipodal pair of separated graphs relevant to v. So by Proposition 3.4.5, there exists $C_i \in G_i$, i=1,2 s.t. $C_1 \Leftrightarrow C_2$. Let $x_1 \in W(G_1)-W(G_2)$, $x_2 \in W(G_2)-W(G_1)$, $y_1 \in C_1-C$, and $y_2 \in C_2-C$.

Subcase 4(a): There exists an incompatible pair (G_3,G_4) s.t. $W(G_4)\neq\{v\}$.

Suppose $G_i|G_j$ for i,j s.t. i=1,2 and j=3,4. Then using Lemma 3.5.4 it be seen that G will be isomorphic to one of H_{14} to H_{17}, H_{21},H_{22},H_{25}, I H_{32} to H_{34} if $G_3 \sim G_4$; otherwise, G will be isomorphic to one of the -phs H^{*}_{35}, H^{**}_{35}, H^{**}_{28}, H^{*}_{31} and H_{36}.

Assume, wlg, that G_1 is attached to G_3. If $|W(G_3)| \geq 2$, then $G_3 = G_1$ $G \in \mathcal{F}_{\mathcal{G}_1}$. In this case G will be isomorphic to one of the graphs H_{37}, , H^{***}_{27}, H_{29}, and H_{35}. Suppose $|W(G_3)|=1$. Then $G_3 \sim G_4$. So G will be morphic to one of $\dot{H}_{12}$, H_{13}, H_{15}, H_{18}, H_{23}, H_{24}, H_{25}, H_{33}, H_{34}, and H_{39}.

case 4(b): There exist G_i, G_j, and G_k s.t. $G_i > G_j$, $G_i > G_k$ and v, v_j v_k are all distinct, where $\{v_j\}= W(G_j)$ and $\{v_k\}= W(G_k)$.

Since $G \in \mathcal{F}_{\mathcal{G}_1}$, $G_1|\ G_i$ and $G_2\ |G_i$. Let $z_i\in C_i-C$, $z_j \in C_j-C$, and $z_k \in$ C, where C_i,C_j, and C_j are some principal cliques of G_i, G_j, and G_k, ectively. Then $G[\{v,x_1,x_2,y_1,y_2,v_i,v_j,v_k,z_i,z_j,z_k\}]$ is isomorphic to . Since $G \in \mathcal{F}_{\mathcal{G}_1}$, G will be isomorphic to H^{***}_{28}.

5: There exist two pairs of antipodal pair of separated graphs.

ase 5(a): There exists G_1,G_2, and G_3 s.t. (G_2,G_1) and (G_2,G_3) are two of incompatible pair.

Now (G_1,G_3) is not an incompatible pair, because then $|W(G_i)|=1$ for i= whence $(G_1 \cup G_2 \cup G_3)-(C-W(G_1)$ will not be a PV-graph. Clearly, | ≥2, and $|W(G_1)|=|W(G_3)|=1$. Since (G_2,G_1) and (G_2,G_3) are two itible pairs, $G_2-(C-W(G_2)$ will contain a subgraph isomorphic to one to H'_9. So G will be isomorphic to one of H_{27}, H_{10}, H_9, and H_{28}.

5(b): There exist G_1, G_2, and G_3 s.t. (G_1,G_2) and (G_3,G_2) are two incompatible pair.

subcase 5(a) (G_1,G_3) is not an incompatible pair. SO $|W(G_1)| \geq 2$, ≥2. Now each of G_1 and G_3 contains a subgraph isomorphic to one H'_5 to H'_9 in Figure 3.5.3. It can be seen than G will

be isomorphic to to a member of S_4.

Subcase 5(c): There exist two pairs (G_1,G_2) and (G_3,G_4) of incompatible pair s.t. $G_i|G_j$ for i=1,2 and j=3,4.

If $|W(G_1)| \geq 2$ and $|W(G_3)| \geq 2$, then it can be seen that G will be isomorphic to a member of S_3. If exactly one of $|W(G_1)|$ and $|W(G_3)|$ is at least 2, then G will be isomorphic to a member of S_2. If $|W(G_i)| = 1$ for all i, i=1,2,3,4, then G will be isomorphic to a member of S_1.

Note that H_{10} is isomorphic to H_{10}^{*} when i=1, and is isomorphic to H_{10}^{**} when i ≥2. H_9 is isomorphic to H_9^{*} if i=0, and is isomorphic to H_9^{**} if i ≥ 1. H_9^{***} is isomorphic to H_9^{**}. H_{27} is isomorphic to H_{27}^{*} if i=1, and is isomorphic to H_{27}^{**} if i ≥ 2. Again H_{27}^{***} is isomorphic to H_{27}. H_{28}^{*} is isomorphic to H_{28}^{**}. H_{28} is isomorphic to H_{28}^{***} if i=0, and is isomorphic to H_{28}^{*} if i ≥ 1. H_{29} is isomorphic to H_{29}^{*} if i=0, and is isomorphic to H_{29}^{**} if i ≥ 1. H_{31}^{*} is isomorphic to H_{31}. H_{35} is isomorphic to H_{35}^{*} if i=0, and is isomorphic to H_{35}^{**} if i ≥ 1.

Hence the necessity is established.▮

Next we give a method for finding out the members of $\mathcal{F}_{\mathcal{T}}$ from that of $\mathcal{F}_{\mathcal{T}_1}$. To do this the only thing we need is that the structure of $G[V(G_1) \cup V(G_2)-(C-W(G_1))]$, where (G_1,G_2) is an incompatible pair of G, $G \in \mathcal{F}_{\mathcal{T}}$, of arbitrary height. Lemma 3.5.4 gives the structure of $G[V(G_1) \cup V(G_2)-(C-W(G_1))]$ if height(G_1,G_2)=1. Assume that we have a method to know the structure of $G[V(G_1) \cup V(G_2)-(C-W(G_1))]$ if height(G_1,G_2) ≤k-1. let $G \in \mathcal{F}_{\mathcal{T}}$, and (G_1,G_2) be an incompatible pair w.r.t. C of height k. Let $W(G_1) \cap W(G_2)= \{v\}$. Let T_1 be a PV-clique tree for G_1, and let $\pi(v)=C,C_1,\ldots,C_i$. Since k ≥ 2, there exists an incompatible pair (G_1',G_2') of G_1 w.r.t. C_1 of height k-1. So, by our assumption, we know the structure of $G[V(G_1') \cup V(G_2')-(C_1-W(G_1'))]$. Now (G_1,G_2) is an incompatible pair. So one of the following two cases holds.

Case 1: $G_1 > G_2$.

Then clearly $G_2-(C-W(G_2)$ will be isomorphic to the complete graph on two vertices. Again, using the similar argument as in Lemma 3.5.4, we can get the structure of $G' = G[V(G_1) \cup V(G_2)-(C-W(G_1))$.

Case 2: $G_1 \sim G_2$.

Wlg, let $depth(G_1,v) \geq depth(G_2,v)$. So it is enough to describe the structure of G_1. Using a similar argument as in Lemma 3.5.4, we can get the structure of G_1. Similarly, we can get the structure of G_2, and hence that of $G[V(G_1) \cup V(G_2)-(C-W(G_1))$.

So we have a method to construct $G[V(G_1) \cup V(G_2)-(C-W(G_1))$, if (G_1,G_2) is an incompatible pair of arbitrary height. Now along the same lines of Theorem 3.5.5, we can find out $\mathcal{F}_{\mathcal{T}}$. So we have the forbidden subgraph characterization for PV-graphs, namely the following:

Theorem 3.5.6: G is a PV-graph iff it does not contain a member of $\mathcal{F}_{\mathcal{T}}$ as an induced subgraph.

3.6: Recognition Algorithm For PV-graphs:

In this section, we present a polynomial time recognition algorithm for PV-graphs. Moreover, if the input graph is a PV-graph, then our algorithm constructs a PV-clique tree T in polynomial time.

Since every chordal graph having at least three cliques has a separating clique, we have the following result.

Theorem 3.6.1: (Atom theorem for PV-graphs) Every atom of a chordal graph is a PV-graph.

Let $X= (G_1,G_2,\ldots,G_r)$ be an ordered set of separated graphs of G w.r.t. C, and $Y=(T_1,T_2,\ldots,T_r)$ be s.t. T_i is a PV-clique tree for G_i, $1 \leq i \leq r$. (X,Y) is said to be perfect w.r.t. C if (1) X contains neither an antipodal pair (G_i,G_j), nor an incompatible pair (G_i,G_j) of separated graphs, (2) There exists no G_i in X s.t. $N(D(G_i)) \geq 2$, (3) The ordering

$G_1, G_2, \ldots, G_r$ satisfy the property of Lemma 3.4.14, and (4) if (G_i, G_j) is a congruent pair with $i < j$, and $C_i \neq C$ is an end vertex of $\pi(v_j)$ in T_i, where $\{v_j\} = W(G_i)$, then $I_{T_i}(C_i) = 0$.

Let T and T′ be any two trees s.t. $v \in V(T)$, and $w \in V(T')$. Define f by $f(T,v,T',w,x) = T''$, where $V(T'') = ((V(T) \cup V(T') \cup \{x\}) - \{v,w\})$, and $E(T'') = (E(T) \cup E(T')) - (\{vv' \text{ s.t. } vv' \in E(T)\} \cup \{ww' \text{ s.t. } ww' \in E(T')\}) \cup \{xx' \text{ s.t. either } vx' \in E(T) \text{ or } wx' \in E(T')\}$. In other words, $f(T,v,T',w,x)$ is the tree T" obtained by merging the vertex v of T with the vertex w of T′, and naming the merged vertex x.

Next we present a procedure 'CONSTRUCT TREE' to construct a tree from a certain collection of trees.

PROCEDURE CONSTRUCT TREE(G_i, T_i, $1 \le i \le r$, C);

INPUT: $X = (G_1, G_2, \ldots, G_r)$, and $Y = (T_1, T_2, \ldots, T_r)$ s.t. (X,Y) is a perfect pair w.r.t. C.

OUTPUT: A PV-clique tree T for $(\bigcup_{i=1}^{r} G_i)$.

METHOD

BEGIN

$T := T_1$;

For $i := 2$ to r do

If G_i is unattached with every G_j $1 \le j \le i-1$ Then

$T := f(T,C,T_i,C,C)$ ELSE

If $C_i \neq C$ is an end vertex of $\pi(v_i)$ in T, where $\{v_i\} = W(G_i)$

Then $T := f(T,C_i,T_i,C,C_i)$

END.

Note that the correctness of the above procedure follows from the proof of the sufficiency of the Theorem 3.4.15. Since, given a PV-clique tree T′ for a graph G′, and a vertex $v \in V(G')$, $\pi(v)$ can be constructed in $O(|V(G')| + |E(G')|)$ time, and since $r = O(n)$, the procedure 'CONSTRUCT TREE'

$O(n(n+m))$ time, where n, and m are the number of vertices and edges, ...tively of $(\bigcup_{i=1}^{r} G_i)$.

...e next suggest a procedure to test whether a given separated graph is ...atible w.r.t. a prescribed vertex.

...URE INCOMP(G_1,C,v,T');

A separated graph G_1 w.r.t. C, a vertex v, $v \in W(G_1)$, and a PV-clique tree T' for G_1.

: If G_1 is incompatible w.r.t. v, then $I(G_1)=0$ Else $I(G_1)=1$, and a PV-clique tree T_1 for G_1 s.t. $I_{T_1}(C_1)=0$, where $C_1 \neq C$ is an end vertex of the path $\pi(v)$ in T_1.

...GIN

...TEP 1: Let $C' \neq C$ be an end vertex of $\pi(v)$ in T';

If $I_{T'}(C') = 0$ Then

begin

$I(G_1)=1$; $T_1:=T'$; GO TO 9

end;

...EP 2: Let $G'_1, G'_2, \ldots, G'_r$ be the separated graphs w.r.t.

C' s.t. $C \in G'_1$;

For i:=1 to r do

$P(G'_i):=G_1$;

If either (i) there exists G'_i s.t. G'_i is attached to G'_1, or (ii) there exist G'_i, G'_j s.t. $G'_i \Leftrightarrow G'_j$, or (iii) there exists G'_i s.t. $N(D(G'_i)) = 2$ Then

begin

STEP 3: (i) For $i:=1$ to r do

begin

Construct a PV-clique tree T'_i for G'_i from T';

$I(G'_i):=1$;

end;

(ii) For every pair (G'_i, G'_j) s.t. $G'_i > G'_j$

with $\{v'_j\}=W(G'_j)$, $1 \le i < j \le r$ do

begin

INCOMP(G'_i, C', v_j, T'_i);

If $I(G'_i)=0$ Then $I(P(G'_i))=0$ and GO TO 9;

end;

(iii) For every congruent pair(G'_i, G'_j), $1 \le i,j \le r$,

and $\{v'_j\}=W(G'_i)$ do

begin

INCOMP(G'_i, C', v'_j, T'_i);

INCOMP(G'_j, C', v'_j, T'_j);

If $(I(G'_i)=0$ and $I(G'_j)=0)$ Then

$I(P(G'_i)=0$ and GO TO 9;

end;

STEP 4: Sort the G'_i s according to lexicographically non-increasing order of ($|W(G'_i)|$, $I(G'_i)$);

Let $X=(G_1^*, G_2^*, \ldots, G_r^*)$ be the new ordering of the separated graphs. Let $Y=(T_1^*, T_2^*, \ldots, T_r^*)$ be s.t. T_i^* is a PV-clique tree for G_i^* constructed previous to this point of time;

CONSTRUCT TREE$(G_i^*, T_i^*, 1 \le i \le r, C')$;

9: STOP;

END.

Lemma 3.6.2: If G'_1 and G'_i are as in the condition (i) of Step 2 of the procedure INCOMP, then G_1 is incompatible w.r.t. v.

Proof: If $|W(G'_1)|\geq 2$, then (G'_i,G'_j) is an antipodal pair relevant to a vertex v_1, $v_1\neq v$, w.r.t. C'. So, by Lemma 3.5.1, G_1 is incompatible w.r.t. v. Assume that $|W(G'_i)|=1$. Let $v_1 \in W(G'_1) \cap W(G'_i)$. Let T_1 be any PV-clique tree for G_1 s.t. C' is an end vertex $\pi(v)$. Since $G'_i > G'_i$, by Propositions 3.4.9 and 3.4.11, either the subtrees corresponding to G'_i and G'_i lie in different branches of C' or all the relevant cliques of G'_1 w.r.t. C' containing v_1 lie in the path $\Pi(C',C'_1)$ of T_1, where C'_i is a principal clique of G'_i w.r.t. C'. Since G_1 is a PV-graph and $\pi(v)$ contains at least three vertices of T_1, the subtrees corresponding to G'_i and G'_j lie in different branches of C'. So $I_{T_1}(C')=1$. So G_1 is incompatible w.r.t. v.∎

To analyze the time complexity of procedure INCOMP we need to construct a tree $T^{(I)}(G_1)$ iteratively, which we call an INCOMP clique decomposition tree for G_1. Let G_1,T_1,C, and v be some input of the procedure INCOMP. Let $C'\neq C$ be an end vertex of $\pi(v)$ in T_1. If $I_{T_1}(C')=0$, then $T^{(I)}(G_1)$ is the tree on the vertex set $\{C,G_1\}$. Let $I_{T_1}(C')=1$. Let G'_i be the separated graph of G_1 w.r.t. C', and T'_i be a PV-clique tree for G_i, $1 \leq i\leq r$, constructed from T_1. If any of the conditions of Step 2 of the Procedure INCOMP is violated, then $T^{(I)}(G'_i)$ is the tree on the vertex set $\{C', G'_i\}$, $1 \leq i \leq r$. If either (i) G'_i is unattached with all other G'_j, or (ii) G'_i is attached to G'_j implies $I_{T'_i}(C'_i)=0$, where $C'_i\neq C'$ is an end vertex of $\pi(v'_j)$, $\{v'_j\} =W(G'_i) \cap W(G'_j)$, then $T(G_i^{(I)}$ is the tree on the vertex set $\{C',G'_i\}$; otherwise, let $T^{(I)}(G'_i)$ be an INCOMP clique decomposition tree for G'_i. Now construct $T^{(I)}(G_1)$ from $T^{(I)}(G'_i)$, $1\leq i\leq r$, by merging the vertex C' of each of $T^{(I)}(G'_i)$ and then adding a new vertex C and joining the edge CC'. We define the root of $T^{(I)}(G_1)$ to be C.

Note that Max { $|V(T^{(I)}(G_1))|$ s.t. G_1 is a separated graph of a chordal graph G having n vertices } is O(n).

Theorem 3.6.3: Procedure INCOMP is correct and can be implemented in O(nm) time, where n and m are the number of vertices and edges of the input graph.

Proof: The correctness of procedure INCOMP follows from Lemma 3.5.1, Lemma 3.6.2, and the correctness of procedure CONSTRUCT TREE.

Given a PV-clique tree T for a PV-graph, a PV-clique tree T_1 for a separated graph G_1 of G can be constructed in O(n+m) time. Since the number of vertices of $T^{(I)}(G_1)$is at most O(n), at most O(nm) time is needed for constructing PV-clique trees for all the separated graphs constructed throughout the procedure INCOMP, from the input tree T'. For the similar reason at most O(nm) time is needed to implement Step 2. Step 4 takes at most O(nm) time in total. We need $O(n^2)$ time to check the conditions in Step 3. So procedure INCOMP takes O(nm) time.∎

Next we present a procedure which will be used in our main algorithm.

PROCEDURE TREE(C);

INPUT: The set $X=(G_1,G_2,...,G_r)$ of all separated graphs of a chordal graph G w.r.t. C, and a set $Y=(T_1,T_2,...,T_r)$ s.t. T_i is a PV-clique tree for G_i, $1 \le i \le r$.

OUTPUT: If G is a PV-graph then output 'TEST(C)=1, and a PV-clique tree T for G. Otherwise, output 'TEST(C)=0'.

METHOD

BEGIN

STEP 1: TEST(C):=1;

If the separated graphs violate any of the conditions of the Theorem 3.16 Then TEST(C):=0;

GO TO 9;

STEP 2: $Z := \phi$;

If there exists an antipodal pair (G_1, G_2) Then

$Z := \{G_2\}$; $X := X - \{G_2\}$ ELSE

If either (i) there exists an incompatible pair (G_i, G_j), or (ii) there exists G_i s.t. $N(D(G_i)=2$

Then $Z := \{G_i\}$; $X := X - \{G_i\}$;

STEP 3: (i) Sort the elements of X in non-increasing order of $|W(G_j)|$, $G_j \in X$;

Let the ordering be $G_1^*, G_2^*, \ldots, G_r^*$, where t=r or r-1 depending on whether $Z = \phi$ or not.

(ii) For i:= 1 to t do

$I(G_i^*)=1$;

(iii) For every pair (G_i^*, G_j^*) with $i<j$ and $W(G_i^*) \cap W(G_j^*) \neq \phi$ do

$INCOMP(G_i^*, T_i^*, v_j, C)$, where T_i^* be the PV-clique tree for G_i^*, and $\{v_j\} = W(G_i^*) \cap W(G_j^*)$.

(iv) Sort the G_i^* s according to lexicographically non-increasing order of $(|W(G_i^*)|, I(G_i^*))$.

Let the new ordering be $G'_1, G'_2, \ldots, G'_t$, and let T'_i, be a PV-clique tree for G'_i, $1 \le i \le t$;

CONSTRUCT TREE$(G'_i, T'_i, 1 \le i \le t, C)$;

Let this tree be $T^{(1)}$; If $Z = \phi$ Then $T := T^{(1)}$ Else $T := f(T^{(1)}, C, T_i, C, C)$;

9: Stop;

END.

Since the procedure CONSTRUCT TREE and the procedure INCOMP are true, the correctness of procedure TREE follows from Theorem 3.4.15. Note that procedure TREE takes at most $O(n^3m)$ time.

Let G be a chordal graph and C be a separating clique of G. Let C separate G into $G[V_i \cup C]$, $1 \le i \le r$, $r \ge 2$. By repeating this process we obtain a clique decomposition of G. This process can be represented by a clique decomposition tree associating each leaf vertex with an atom of G and each internal vertex with a clique separator of G. The original graph can be reconstructed by composing subgraphs in the decomposition tree. This Clique decomposition of a chordal graph can be done in O(nm) time (see[92]).

ALGORITHM A:

INPUT: A graph G.

OUTPUT: A PV-clique tree T for G iff G is a PV-graph; otherwise, output 'G is not a PV-graph'.

METHOD

BEGIN

STEP 1: If G is not a chordal graph then output 'G is not a PV-graph'.

STEP 2: Construct a clique decomposition tree $T^{(S)}$ for G.

STEP 3: Construct PV-clique trees for each Atom.

STEP 4: Let the root of $T^{(S)}$ be C. Let Max {d(C,C') s.t. C' is not a pendant vertex of $T^{(S)}$}=k, where d(C,C') is the distance from C to C' in $T^{(S)}$; Let S_j={C" $\in$ $V(T^{(S)})$ and C" is not a pendant vertex s.t. d(C,C")=j};

Let $|S_j|=t_j$; Let $C_{j_1}, C_{j_2}, \ldots, C_{j_{t_j}}$ be some ordering of S_j;

For i:=k down to 0 do

For j:=1 to t_i do

begin

TREE(C_{i_j});

If Test(C_{i_j})=0 Then output 'G is not a PV-graph

and GO TO 9;

end;

9: STOP;

END.

The correctness of ALGORITHM A follows from the correctness of the procedure TREE, Theorem 3.4.15, and Theorem 3.6.1. Since the number of separating clique of G is O(n), ALGORITHM A takes $O(n^4m)$ time.

From the above we have the following Theorem.

Theorem 3.6.4: PV-graphs can be recognized in $O(n^4m)$ time. Moreover, a PV-clique tree T for a PV-graph G can be constructed in $O(n^4m)$ time.

Though we have a polynomial recognition algorithm for PV-graphs, it would be interesting to design a more efficient recognition algorithm for PV-graphs.

We have seen in Theorem 3.2.1 that the class of intersection graphs of edge disjoint paths in a tree and the class of intersection graphs of edge disjoint subtrees in a tree are one and the same. Since the graph H_4 in Figure 3.5.1 is not a PV-graph but it can be seen that it is an intersection graphs of vertex disjoint subtrees of a tree, it would be interesting to characterize intersection graphs of vertex disjoint subtrees in a tree following the frame work of Monma and Wei[92]. We left the problem of obtaining the forbidden subgraphs for this class using the framework developed in Chapter 2 as an open problem.

Let G and H be two vertex disjoint graphs and let x be a vertex of G. By substituting H for x, we mean deleting x and joining every vertex of H to those vertices of G which were adjacent to x. Let $G_1=(V_1,E_1)$ and $G_2=(V_2,E_2)$ be any two graphs s.t. $V_1=\{x_1,x_2,\ldots x_{n_1}\}$ and $V_2=\{y_1,y_2,\ldots,y_{n_2}\}$. Let $C_1=\{x_1,x_2,\ldots,x_r\}$ and $C_2=\{y_1,y_2,\ldots,y_r\}$ be any cliques of G_1 and G_2 respectively. Then the clique bonding of G_1 and G_2 is another graph G_3 which is defined as follows:

$V(G_3)=\{x_{r+1},x_{r+2},\ldots,x_{n_1},y_{r+1},y_{r+2},\ldots,y_{n_2},\ z_1,z_2,\ldots,z_r\}$, and $E(G_3)=\{x_ix_j$ if $x_ix_j\in E(G_1)$, $r+1\le i,j\le n_1\}\cup\{y_iy_j$ s.t. $y_iy_j\in E(G_2)$, and $r+1\le i,j\le n_2\}\cup\{z_iz_j$ s.t. $1\le i,j\le r\}\cup\{z_ix_j$ s.t. $x_ix_j\in E(G_1)$ and $1\le i\le r$, $r+1\le j\le n_1\}\cup\{z_iy_j$ s.t. $y_iy_j\in E(G_2)$ and $1\le i\le r$, $r+1\le j\le n_2\}$.

Theorem 4.2.2[86]: The graph obtained by substituting a perfect graph for some vertex of a perfect graph is a perfect graph.

Proof of Theorem 4.2.1: Assume that $\alpha=(u\ v)$ is a transposition of $\Gamma(G)$, $G\in \mathcal{F}_S$. Let $H_1=G-u$. Since $\Gamma(G)=\{\beta$ s.t. $xy\in E(G)$ iff $\beta(x)\beta(y)\in E(G)\}$, and $\alpha\in\Gamma(G)$, we have $N(u)=N(v)$ if $uv\notin E(G)$; otherwise, $N[u]=N[v]$. If $uv\notin E(G)$, then G is obtained by substituting a $\bar{K}_2$ to the vertex v of the perfect graph H_1, otherwise G is obtained from H_1 by substituting a K_2 for v of H_1. So by Theorem 4.2.2, G is a perfect graph, which is absurd.▮

Let $\mathcal{S}=\{G$ s.t. $\Gamma(G)$ has a transposition$\}$. Next we show that it is enough to study certain proper subclass of $\mathcal{S}$ as far as SPGC is concerned.

Theorem 4.2.3: Let $\mathcal{S}_1=\{G\in\mathcal{S}$ s.t. G is regular$\}$. Then $\mathcal{S}_1$ is a complete class for SPGC.

Proof: Given a graph G we construct a k-regular graph G′ s.t. $\Gamma(G')$ has a transposition, where $k=2\Delta(G)-1$, and $\Delta(G)$ is the maximum degree of G. We first construct the graph T_k as follows:

Take the complement of the union of one P_3 and $\Delta-1$ copies of K_2, and then take a new vertex and join it to the middle vertex of the original P_3. Now T_k has degree sequence 1, $2\Delta-1$, $2\Delta-1,\ldots,2\Delta-1$. We now clique bond $(2\Delta-1-\deg(x))$ copies of T_k to the vertex x where $\deg(x)$ is the degree of x in G. The bonding is done with the vertex of degree 1 in T_k. The graph G′ so formed is regular of degree $k=2\Delta-1$. Now clearly T_k is a perfect graph. So G′ is perfect iff G is perfect. Again $\Gamma(G')$ has a transposition, namely $\alpha=(u\ v)$, where u and v are the end vertices of some K_2 in the complement of some T_k in G′.∎

Note that $\beta=(x\ y)$, where x and y are the end vertices of the P_3 in the complement of some T_k, is also a transposition of $\Gamma(G')$ s.t. $xy \in E(G')$. Since $\Gamma(G')$ contains a transposition $\alpha=(u\ v)$ s.t. $uv \notin E(G')$ as well as a transposition $\beta=(x\ y)$ s.t. $xy \in E(G')$, we have the following stronger result.

Theorem 4.1.4: Let $\mathscr{S}_2=$ { G s.t. G is regular, and $\Gamma(G)$ contains a transposition $\alpha=(x\ y)$ s.t. $xy \in E(G)$} and $\mathscr{S}_3=${ G s.t. G is regular, and $\Gamma(G)$ contains a transposition $\beta=(x\ y)$ s.t. $xy \notin E(G)$}. Then $\mathscr{S}_2$ and $\mathscr{S}_3$ are complete classes for SPGC.

Next we show a subclass of graphs to be valid for SPGC. The vertex regularity can be viewed as follows: Let $f: V \longrightarrow Z^+$ be defined by $f(v) = |N(v)|$, where $N(v)=$ { w s.t. $vw \in E(G)$}. Now G is regular iff f is constant on V. We define the functions $g: E \longrightarrow Z^+$ by $g(e)=g(uv)=|\{ w$ s.t. $uw \in E(G)$ and $vw \in E(G)\}|$, and $h : \bar{E} \longrightarrow Z^+$ by $h(xy) = |\{z$ s.t. $xz \in E(G)$ and $yz \in E(G)\}|$.

Let $\mathscr{S}_4=$ { G s.t. $G \in \mathscr{S}_1$, and g and h are constant on E, and $\bar{E}$, respectively}. Next we show that $\mathscr{S}_4$ is a valid class for SPGC.

Theorem 4.2.5: $\mathscr{S}_4$ is a valid class for SPGC.

Proof: In fact, we prove that $G \in \mathscr{S}_4$ implies that G is perfect. Wlg, Let G

$\in \mathcal{S}_4$ be a connected r regular graph on n vertices. Let $(u\ v) \in \Gamma(G)$.

Case 1: $uv \in E(G)$.

Now $N[u]=N[v]$. Again $g(uv)=r-1$. Since g is constant on E, $g(xy)=r-1$ for every $xy \in E(G)$. Let $w \in N(u)$, $w \neq v$. Since $g(uw)$ r-1, and $\deg(u)=r$, so $N[u]=N[w]$. Since $N[u]=N[w]$ for every $w \in N(u)$, and since G is connected, G is isomorphic to K_{r+1}, whence G is perfect.

Case 2: $uv \notin E(G)$.

Now $N(u)= N(v)$. Again $h(uv)=r$. Since h is constant on $\bar{E}$, $h(xy)= r$ for every $xy \notin E(G)$. Consider $\bar{G}$. Now $\bar{G}$ is regular and $\Gamma(G)= \Gamma(\bar{G})$. So $\Gamma(\bar{G})$ contains a transposition $(u\ v)$ s.t. $uv \in E(\bar{G})$. Again g is constant on $E(\bar{G})$, and $g(uv) = (n-r-2)$, for every $uv \notin E(G)$. So by case 1, every connected component of $\bar{G}$ is isomorphic to $K_{(n-r)}$. So $\bar{G}$ is perfect, whence G is perfect.█

Actually we have proved the following stronger result.

Theorem 4.2.6: The class $\mathcal{S}_5$={G $\in \mathcal{S}$, and g is constant on E} and $\mathcal{S}_6$= {G $\in \mathcal{S}$, and h is constant on $\bar{E}$} are valid classes for SPGC.

4.3 Stability of Graphs:

Though stability of graphs is defined in chapter 1, we recall the definition.

A graph G of order n is said to be stable if there is a sequence $\alpha=(v_1, v_2,\dots,v_n)$ of V s.t. $\Gamma(G_{S_j})= \Gamma(G)_{S_j}$, $1 \le j \le n$, where $S_j=\{v_1, v_2,\dots,v_j\}$, and $G_{S_j} = G-S_j$. α is then said to be a stabilizing sequence.

The notion of stability was introduced by Holton[69]. Stability of graphs has been extensively studied[69-72,125]. A necessary condition for a graph G to be stable is that $\Gamma(G)$ contains a transposition. Unfortunately, there is no characterization for stability of graphs. G is said to be semi stable at v if $\Gamma(G_v)= \Gamma(G)_v$. In [69] Holton proved the

following:

If $\Gamma(G) \leq D_n$, where D_n denotes the dihedral group , $n \geq 5$, then G is unstable.

From above it is clear that the automorphism group of a graph plays an important role in deciding the stability of a graph. However, this is not the only factor, because there are graphs having same automorphism groups which are not simultaneously stable[69]. This might have motivated Holton for the following two conjectures(see [69, pp 166]).

Conjecture (1): If G, H, $\bar{G}$, and $\bar{H}$ are all connected and $\Gamma(G)=\Gamma(H)$, then G is stable iff H is stable.

Conjecture (2): If G and H are connected but $\bar{G}$ and $\bar{H}$ are both disconnected and $\Gamma(G)=\Gamma(H)$, then G is stable iff H is stable.

We first show that conjecture (2) implies conjecture (1). Then we disprove conjecture (1) by producing an infinite class of counter examples in the class of perfect graphs, hence disprove conjecture (2) also.

To this end we need the following results.

Theorem 4.3.1 [69]: $\bigcup_{i=1}^{n} G_i$ is stable iff each G_i is stable.

Theorem 4.3.2 [67]: $\Gamma(G_1 \cup G_2)= \Gamma(G_1) + \Gamma(G_2)$ iff no component of G_1 is isomorphic to any component of G_2.

Theorem 4.3.3 [69]: G is stable iff $\bar{G}$ is stable.

Theorem 4.3.4 [72]: G is semistable at v iff N(v) is fixed by $\Gamma(G_v)$.

Theorem 4.3.5 [70]: If G is stable, then $\Gamma(G)$ has a transposition.

Theorem 4.3.6 [71]: A tree T is stable iff $\Gamma(T)$ has a transposition.

We next show that conjecture 2 implies conjecture 1.

Theorem 4.3.7: If Conjecture (2) is true, then conjecture (1) is true.

Proof: If not, there exist G and H s.t. G, H, $\bar{G}$, and $\bar{H}$ are all connected, $\Gamma(G)= \Gamma(H)$, G is stable but H is unstable. Let $G_1= \overline{G \cup K_1}$, and $H_1= \overline{H \cup K_1}$.

It is easy to check that G_1 and H_1 satisfy all the conditions of conjecture (2). since $\bar{G}_1 = G \cup K_1$ is stable by Theorem 4.3.1, G_1 is stable by Theorem 4.3.3. Similarly H_1 is unstable because H is unstable, implying the falsity of Conjecture (2).∎

In view of Theorem 4.3.7, if we could show that conjecture (1) is false, then conjecture (2) would be false. The following Theorem gives an infinite class of counter examples for conjecture (1).

Theorem 4.3.8: The graphs G_n and H_n given in Figure 4.3.1 satisfy the following:

(a) G_n, H_n, $\bar{G}_n$, and $\bar{H}_n$ are all connected.

(b) $\Gamma(G_n) = \Gamma(H_n) = S_2(2) + S_2 + I_n$.

(c) G_n is stable but H_n is unstable.

Proof: (a) It is easy to check that G_n, H_n, $\bar{G}_n$, and $\bar{H}_n$ are all connected.

(b) The labellings of G_n and H_n show that $\alpha \in \Gamma(G_n)$ iff $\alpha \in \Gamma(H_n)$. It is easy to check that $\Gamma(G_n)=\Gamma(H_n)= S_2(2)+S_2+I_n$.

(c) It is easy to check that G_n is semistable at a. Now $G'_n = G_n - a$ is a tree. Since $\Gamma(G'_n)$ has a transposition (e f), G'_n is stable by Theorem 4.3.6. So G_n is stable. We claim that H_n is unstable. One can verify using Theorem 4.3.4 that H_n is semistable at x iff $x \in \{ e, f, v_1, v_2, \ldots, v_n\} - \{v_{n-2}\}$. If x = e or x=f, then $\Gamma(H_n - x)$ does not contain any transposition. So $H_n - x$ is unstable by Theorem 4.3.5. Assume that $x \in \{ v_1, v_2, \ldots, v_n\} - \{ v_{n-2}\}$. Now x is a cut vertex of H_n. Let H'_n be the connected component of $H_n - x$ containing a. Then $\Gamma(H'_n)$ does not contain any transposition. Hence by Theorem 4.3.5, H'_n is unstable. So by Theorem 4.3.1, H'_n is unstable. Since H_n is semistable at v implies that $v \in \{ e, f, v_1, v_2, \ldots, v_n\} - \{v_{n-2}\}$, H_n is not stable. Hence our claim is established.∎

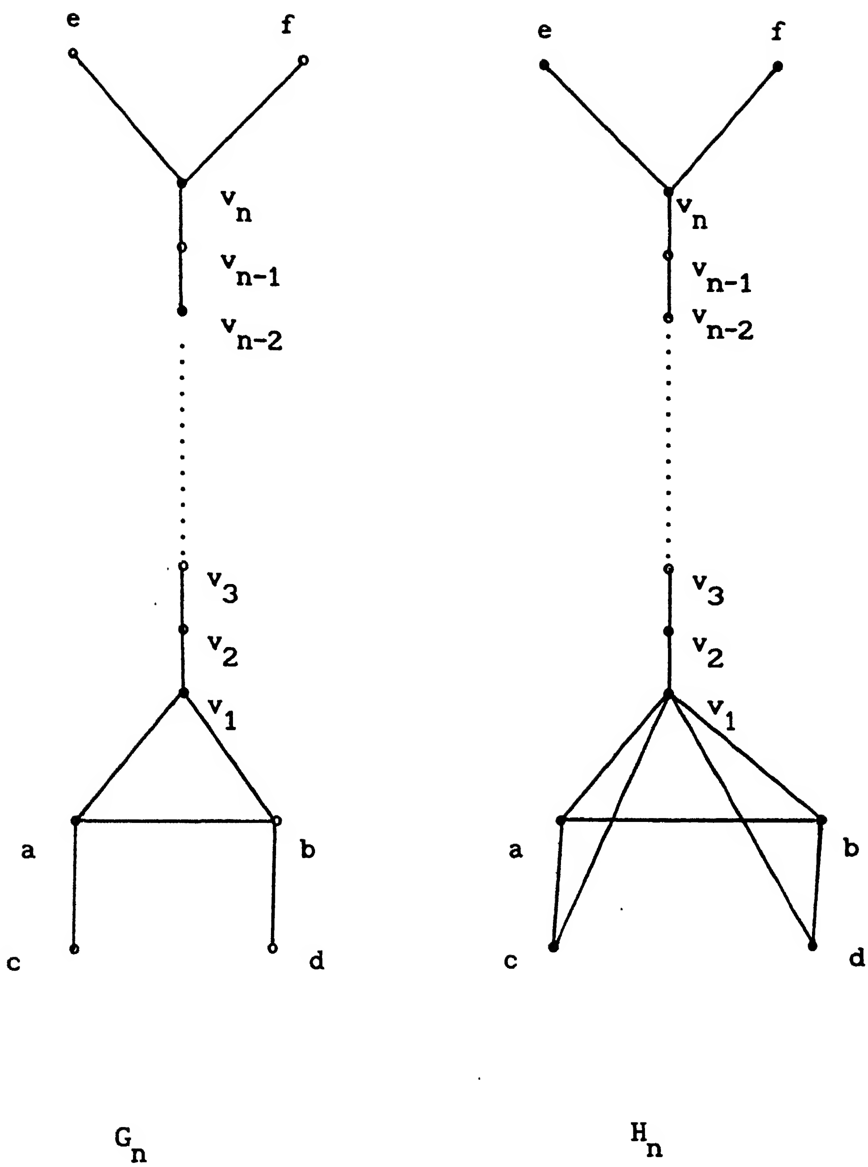


Figure 4.3.1: A Counter example to Conjecture (1).

Using Theorem 4.3.7, and Theorem 4.3.8, one can construct an infinite class of counter examples for conjecture (2). Next we construct another infinite class of counter examples for Conjecture (2).

We first show that if one counter example for conjecture (2) is known, then we can construct an infinite class of counter examples.

Theorem 4.3.9: Let G and H be two connected graphs s.t. $\overline{G}$ and $\overline{H}$ are disconnected, $\Gamma(G)= \Gamma(H)$, G is stable but H is unstable. Define G_n and H_n recursively as, $G_1=G$, $H_1=H$, and $G_n= \overline{G_{n-1}\cup K_1}$, $H_n=\overline{H_{n-1}\cup K_1}$, $n>1$. Then G_n and H_n are both connected, $\overline{G}_n$ and $\overline{H}_n$ are both disconnected, $\Gamma(G_n)=\Gamma(H_n)$ and G_n is stable but H_n is unstable for all $n \geq 1$.

Proof: It follows easily from induction that $\overline{G}_n$ and $\overline{H}_n$ are disconnected. Hence G_n and H_n are connected for all $n \geq 1$. Now $\Gamma(G_n)=\Gamma(\overline{G}_n)= \Gamma(G_{n-1}+K_1)= \Gamma(G_{n-1})+\Gamma(K_1)=\Gamma(H_{n-1})+\Gamma(K_1)=\Gamma(\overline{H}_n)=\Gamma(H_n)$. So G_n and H_n have same automorphism groups for all $n \geq 1$.

Since $\overline{G}_n$ is the union of two stable graphs, $\overline{G}_n$ is stable. Hence G_n is stable. Now $\overline{H}_n$ is the union of a stable graph and an unstable graph. So $\overline{H}_n$ is unstable. Hence H_n is unstable.▌

Note that in Theorem 4.3.9, in place of K_1, one can take any stable graph none of whose components is isomorphic to any component of H.

The following Theorem gives a counter example for Conjecture (2).

Theorem 4.3.10: The pair of graphs (G,H), where $\overline{G}$ and $\overline{H}$ are given in Figure 4.3.2, is a counter example for conjecture (2).

Proof: It is easy to check that, G and H are both connected, $\overline{G}$, and $\overline{H}$ are both disconnected, and $\Gamma(G)= S_2(2)+S_2+S_1= \Gamma(H)$. G is a stable graph with a stabilizing sequence $\alpha=(1,2,3,4,5.6,7)$. As the connected component $\overline{H}_1= \overline{H}[\{v_2,v_3,v_4,v_5,v_6\}]$ of $\overline{H}$ does not have any transposition in its automorphism group, $\overline{H}_1$ is unstable by Theorem 4.3.5. Hence $\overline{H}$ is unstable by Theorem 4.3.1. Thus H is unstable by Theorem 4.3.3.▌

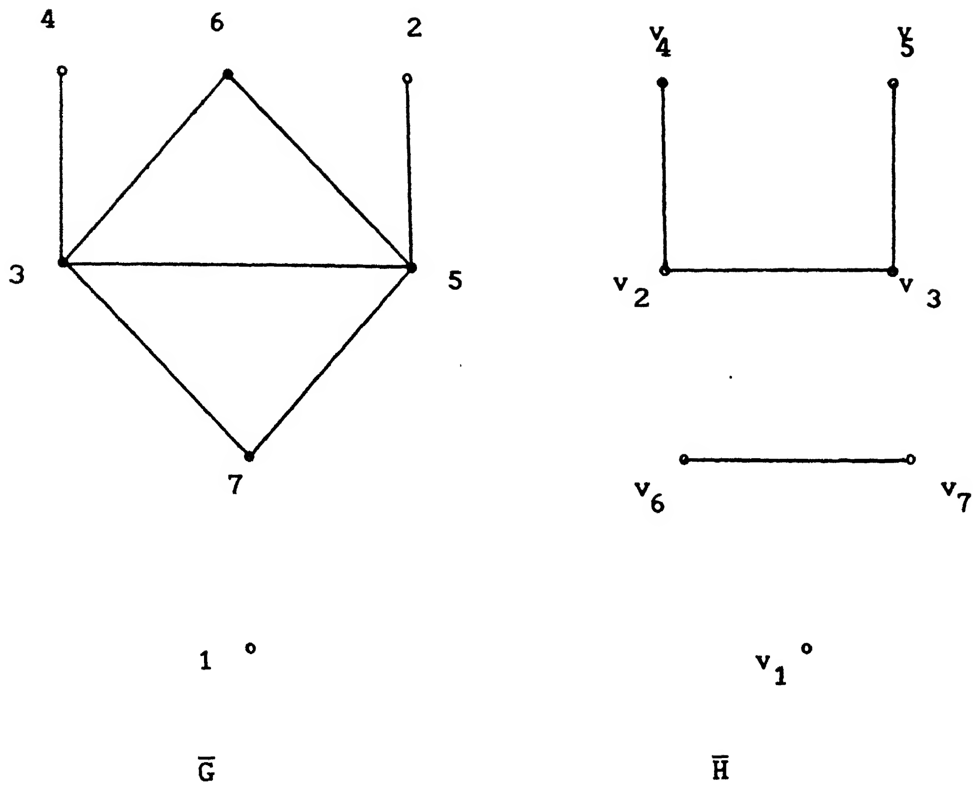


Figure 4.3.2: A Counter example to conjecture (2).

CHAPTER 5
PERFECT ELIMINATION ORDERINGS IN CHORDAL GRAPHS

5.1 Introduction:

Perfect elimination ordering (PEO) plays an important role in obtaining structural results and designing efficient algorithms in chordal graphs. Efficient algorithms exist for finding maximum independent sets, maximal cliques, minimum coloring and minimum clique cover in chordal graphs (see Gavril [49]), even though these problems are known to be NP-Hard for general graphs (see [47]). All these algorithms need in their inputs a chordal graph and a PEO. Chordal graphs can be recognized using the following two steps.

Step 1: Compute an ordering α of G that is a PEO iff G is chordal.

Step 2: Test whether α is a PEO. Declare G chordal if α is a PEO.

So there is a lot of interest in PEOs of chordal graphs [49,50,57,76,111,112,114,119,122,123,128-130], and various algorithms have been suggested to generate them [57,114,119,128,129]. BFS can be used to generate PEOs of chordal graphs, namely LEX-BFS algorithms of Rose et al [114] is based on certain lexicographic ordering and it turns out that the search is a BFS. However, it is not known whether DFS can be used to generate PEOs of chordal graphs.

In section 2, we suggest three algorithms, namely (i) Maximum Cardinality breadth first search (MCBFS), (ii) Maximum cardinality depth first search (MCDFS), and (iii) Local maximum cardinality search (LMCS) to generate PEOs of chordal graphs. We will show that MCBFS and MCDFS are natural applications of BFS and DFS, respectively. We will show that MCS, MCBFS, and MCDFS are special cases of LMCS and that MCBFS and MCDFS run in linear time, while LMCS takes $O(n^2)$ time.

It is known [57] that neither MCS nor LEX-BFS can generate any arbitrary PEO of an arbitrary chordal graph. It is also established that there are MCS orderings that cannot be generated by LEX-BFS and conversely [57]. In the same spirit we have shown in section 3 that the algorithms we have suggested in section 2 are different from MCS and LEX-BFS, and none of them can generate any arbitrary PEO of a general chordal graph. we then give a comparative study of these algorithms as far as generating any arbitrary PEO of an arbitrary chordal graph is concerned.

In section 4, we study HEO of chordal graphs. We characterize H-Perfect k-trees in terms of the number of their simplicial vertices, and also in terms of forbidden subgraphs. We also characterize K-trees with a given jump number and suggest a linear time algorithm for generating a jump sequence of an arbitrary K-tree. We then present an $O(n^2m)$ algorithm to recognize H-Perfect chordal graphs. We introduce the notion of extremal chordal graphs and present two characterizations of these graphs and indicate a polynomial recognition algorithm.

In section 5, we characterize Hamiltonian H-Perfect chordal graphs and present a linear time algorithm for finding a Hamiltonian cycle of an H-Perfect chordal graph if an HEO is given in the input.

We present in section 6, a linear time algorithm for finding out a Hamiltonian cycle in a biconnected proper interval graph.

5.2 The Algorithms:

In this section we present three algorithms for generating PEOs of chordal graphs.

Algorithm LMCS:

INPUT: A graph G.

OUTPUT: An ordering $\alpha = (\alpha(1), \alpha(2), \ldots, \alpha(n))$ of V s.t. G is a chordal graph iff α is a PEO of G.

METHOD:

```
BEGIN
 For all v ∈ V do
 L(v):=0;
 Choose an arbitrary vertex x; S:=φ;
 For j:=n  Down to 1 do
 begin
   v:= x; α(j):=v; L(v)= -1; S := S ∪ {v};
   For all w ∈ N(v) do
   If L(w) ≥ 0 then L(w):= L(w)+1;
10 If S = φ then Stop
   Else choose a vertex y ∈ S;
   Choose a vertex y' ∈ N(y) s.t.L(y')=Max{L(z), z∈ N(y)};
   If L(y') > 0 then x := y' Else
   begin
    S:= S - { y}; GO TO 10;
   end;
 end;
END.
```

Now we show that Algorithm LMCS runs in $O(n^2)$ time.

Theorem 5.2.1: Algorithm LMCS takes $O(n^2)$ time.

Proof: Throughout the algorithm every vertex is added to S exactly once and then some vertices are deleted from S. Since each addition of a vertex to S takes at most O(n) time and each deletion of a vertex from S takes at most O(n) time, Algorithm LMCS takes $O(n^2)$ time.∎

Next we present an algorithm based on BFS.

Algorithm MCBFS:

INPUT: A graph G.

OUTPUT: An ordering $\alpha=(\alpha(1), \alpha(2),\ldots,\alpha(n))$ of $V(G)$ s.t. α is a PEO of G iff G is a chordal graph.

METHOD:

```
    BEGIN
     For all v ∈ V do
     L(v):=0;
     Choose an arbitrary vertex x ∈ V;
     L(x):= -1; Create Queue(Q);
     For i:=n Down To 1 do
     begin
       v:=x; α(i):=v; Add(Q,v);
       For all w ∈ N(v) do
       If L(w) ≥ 0 then L(w):= L(w)+1;
10     If empty(Q) then Stop Else
       y:= Front(Q);
       choose a vertex y'∈ N(y) s.t. L(y') = Max {L(z), z ∈ N(y)};
       If L(y') > 0 then x:=y'; L(y'):= -1 Else
       begin
         Delete(Q); GO TO 10;
       end;
     end;
    END.
```

That Algorithm MCBFS runs in linear time is shown in the following theorem.

Theorem 5.2.2: Algorithm MCBFS takes $O(n+m)$ time.

Proof: We implement MCBFS as follows:

We maintain an array of Sets(i) for $0 \le i \le n-1$ for the front vertex of the queue Q. we store in Set(i) all unnumbered vertices which are adjacent

to the front vertex of Q and have L(.) = i. Initially the front of Q contains v and Set(0) contains all the vertices adjacent to v. we maintain the largest index j s.t. Set(j) is non-empty. To carry out a step of the search, we remove a vertex x from Set(j), number it, and enqueue it to Q. For each unnumbered vertex y in N(x), we remove y from Set(L(y)) to Set(L(y)+1) and update L(y) to L(y)+1. Then we add 1 to j, and while Set(j) is empty repeatedly decrement j. j may go down up to -1. Since deg(x) $\geq$ j-1, we charge the cost of manipulating j to x. since no vertex is charged more than once, the total time for manipulating j is $O(\sum \deg(x)) = O(n+m)$ time. Again when a vertex is removed from the front of the queue we have to maintain the array Set(i) for the new front vertex say x′ of Q. So we have to spend deg(x′) time for creating the Set array for x′. So the total time for creating the array Set for all the vertices is O(n+m). If we replace each set by a doubly linked list of vertices (to facilitate deletion), the Algorithm MCBFS takes O(n+m) time.∎

Finally we present an algorithm based on DFS to generate PEOs of chordal graphs.

Algorithm MCDFS:

INPUT: A graph G.

OUTPUT: An ordering $\alpha=(\alpha(1), \alpha(2), \ldots, \alpha(n))$ of V(G) s.t. α is a PEO of G iff G is a chordal graph.

METHOD:

BEGIN

For all $v \in V$ do

L(v):=0;

Choose an arbitrary vertex $x \in V$;

L(x):= -1; Create Stack(S);

For i:=n Down To 1 do

```
     begin
      v:=x; α(i):=v; Add(S,v);
      For all w ∈ N(v) do
      If L(w) ≥ 0 then L(w):= L(w)+1;
10    If empty(S) then Stop Else
      y:= Top(S);
      choose a vertex y' ∈ N(y) s.t. L(y')= Max{ L(z), z ∈ N(y)};
      If L(y') > 0 then x:=y'; L(y'):= -1 Else
      begin
       Delete(S); GO TO 10;
      end;
     end;
    END.
```

Now we show that Algorithm MCDFS runs in linear time.

Theorem 5.2.3: Algorithm MCDFS takes $O(n+m)$ time.

Proof: We implement MCDFS as follows:

We maintain an array L s.t. $L(v)$= the number of numbered vertices adjacent to v, $v \in V$. For the top element v of S we sort the neighbours of v and in the set(v,i) we place the neighbours of v having $L(.)= i$. We maintain the total number of unnumbered neighbours of v and the indices of non-empty sets for v. To do this we have to spend $O(deg(v)$ time. To carry out a step of the search, we find the maximum j of set indices from the list of the set indices of v, and choose a vertex x from the set(v,j), and decrement $T(v)$ by $T(v)-1$, $T(set(v,j))$ by $T(set(v,j)-1$. If set(v,j) is empty, then delete j from the list of set indices of v. To do this we have to spend at most $O(j)$ time. We charge this cost to the vertex x. Then we create set list, sets, $T(x)$, $T(set(x,i))$ for all sets and if $x' \in (N(x) \cap N(v'))$ with $L(v')= -1$, then we delete x' from set$(v',L(x'))$ and decrement

$T(v')$ by $T(v')-1$. So we have to spend at most $O(deg(x))$ time. For the back tracked vertex x we check $T(x)=0$ or not to confirm whether $N(x)$ contains an unnumbered vertex. If $T(x) \neq 0$, then we choose the largest index from the set indices of x. So we have to spend at most $O(deg(x))$ time. we maintain an array CP s.t. $CP(v)$ is the current parent of v. If we store each set using a doubly linked set, Algorithm MCDFS takes $O(n+m)$ time.∎

Next we show that any ordering of a graph G produced by LMCS is a PEO iff G is chordal. To this end we need the following result due to Tarjan et al [129].

Lemma 5.2.4: [129]: Let $G=(V,E)$ be a chordal graph and let α be an ordering of G. If α has the following property, then α is a PEO of G.

(P) If $\alpha(u) < \alpha(v) < \alpha(w)$, $uw \in E(G)$,and $vw \notin E(G)$, then there is a vertex x s.t. $\alpha(v) < \alpha(x)$, $vx \in E(G)$, and $ux \notin E(G)$.

Theorem 5.2.5: Let $G=(V,E)$ be a chordal graph. If α is an ordering produced by LMCS, then α is a PEO of G.

Proof: The only thing we have to show is that α has the property (P) of Lemma 5.2.4. Let u,v, and w be in V s.t. $\alpha(u) < \alpha(v) < \alpha(w)$, $uw \in E(G)$, and $vw \notin E(G)$. Let $\alpha(v)=i$. Since $\alpha(u) < \alpha(v)$,there exists a vertex y s.t. $y \in N(v)$, $\alpha(y) > \alpha(v)$ and $L(v,y)$ is maximum, where $L(v,y)= |\{ w \in N(v) \text{ s.t. } \alpha(w) > i \}|$. If $u \notin N(y)$, then we take y=x, and x is the required vertex. Assume that $u \in N(y)$. Since $\alpha(u) < \alpha(v)$, $L(v,y) \geq L(u,y)$. Again $\alpha(u) < \alpha(w)$ and $uw \in E(G)$ but $vw \notin E(G)$. So there exists a vertex x s.t. $\alpha(x) > i$ and $vx \in E(G)$ and $ux \notin E(G)$. Now x is the required vertex.∎

Note that Algorithms MCBFS and MCDFS follow from Algorithm LMCS by replacing the set S in the LMCS Algorithm by a queue Q, and a stack S, respectively. So we have the following.

Theorem 5.2.6: Let α be an ordering produced by either MCBFS or MCDFS. Then G is chordal iff α is a PEO of G.

5.3 A Comparative Study:

To make our thesis self contained, we shall restate here the details of MCS and LEX-BFS algorithms due to Tarjan et al [129] and Rose et al [114], respectively.

Algorithm MCS:

INPUT: A Graph $G = (V, E)$.

OUTPUT: A numbering $(v_1, v_2, \ldots, v_n)$ of V.

METHOD:

BEGIN

STEP 1: For every $v \in V(G)$ do

MCSL(v):=0;

STEP 2: For i:= n Down To 1 do

Begin

Choose an unnumbered vertex v having largest MCSL(v), ties being broken arbitrarily, and number it by v_i;

For every $w \in N(v)$ do

MSCL(w):=MCSL(w)+1;

End;

END.

Algorithm LEX-BFS:

INPUT: A Graph $G = (V, E)$.

OUTPUT: A numbering $(v_1, v_2, \ldots, v_n)$ of V.

METHOD:

BEGIN

STEP 1: For every $v \in V(G)$ do

$L_v = [0,0,\ldots,0]$, an array of length n s.t. each entry of L_v is 0;

STEP 2: For $i := n$ Down To 1 do

Begin

Choose an unnumbered vertex v having lexicographically largest L_v^{-1}, ties being broken arbitrarily, and number it by v_i;

(* L_v^{-1} $[j]=L_v[n+1-j]$, $1 \le j \le n$ *)

For every $w \in N(v)$ do

$L_w[i]:=1$;

End;

END.

Let $U = [u_1, u_2, \ldots, u_n]$ and $W = [w_1, w_2, \ldots, w_n]$ be two arrays. Then $U \underset{lex}{>} W$ iff either $u_1 > w_1$ or there exists j s.t. $u_i = w_i$ for all $i = 1, 2, \ldots, j$ and $u_{j+1} > w_{j+1}$. $U \underset{lex}{=} W$, iff $u_i = w_i$ for all i, $1 \le i \le n$.

Let $\alpha=(v_1, v_2, \ldots, v_n)$ be an ordering of $V(G)$. For every i,j, $1 \le i < j \le n$, let $DP(i,j)= \min k$, $j \le k$ s.t. $v_i v_k \in E(G)$, and $BP(i,j)= \max k$, $j \le k$ s.t. $v_i v_k \in E(G)$. For every i,j, $1 \le i < j \le n$, let $MPN(v_i,j)= |N(v_i) \cap \{v_j, v_{j+1}, \ldots, v_n\}|$, $LEXPN(v_i,j)= L_{(i,j)}$, the array of length n, s.t. $L_{(i,j)}[k]= 1$ if $k \ge j$ and $v_i v_k \in E(G)$, otherwise $L_{(i,j)}[k]=0$, $1 \le k \le n$, $MCDPN(v_i,j)= [n-DP(i,j), |N(v_{DP(i,j)}) \cap \{ v_j, v_{j+1}, \ldots, v_n\}|]$, $MCBPN(v_i,j)= [BP(i,j), |N(v_{BP(i,j)}) \cap \{ v_j, v_{j+1}, \ldots, v_n\}|]$, and for $i<j \le k$, $LMPN_{v_k}(v_i,j)=$

$$\begin{cases} 0 \text{ if } v_i v_k \notin E(G) \\ |N(v_i) \cap \{v_j, v_{j+1}, \ldots, v_n\}|, \text{otherwise.} \end{cases}$$

Let $LEXPN(v_i,j)^{-1}[k]=LEXPN(v_i,j)[n+1-k]$, $1 \le k \le n$, and for every i,j with $1 \le i < j \le n$.

Any ordering of $V(G)$ of a chordal graph G generated by the MCS (LEX-BFS, MCBFS, MCDFS, and LMCS) algorithm is called an MCS (LEX-BFS, MCBFS, MCDFS, and LMCS) sequence of G.

We have the following easy propositions.

Proposition 5.3.1: Let $\alpha = (v_1, v_2, \ldots v_n)$ be an ordering of $V(G)$ of a connected chordal graph G. Then the following are true.

(1) α is an MCS sequence of G iff $MPN(v_{j-1}, j) \geq MPN(v_i, j)$ for all $1 \leq i < j-1 \leq n-1$.

(2) α is a LEX-BFS sequence iff $LEXPN(v_{j-1}, j)^{-1} \geq_{lex} LEXPN(v_i, j)^{-1}$, for every i, j with $1 \leq i < j-1 \leq n-1$.

(3) α is an MCBFS sequence iff $MCBPN(v_{j-1}, j) \geq_{lex} MCBPN(v_i, j)$, for every i,j with $1 \leq i < j-1 \leq n-1$.

(4) α is an MCDFS sequence iff $MCDPN(v_{j-1}, j) \geq_{lex} MCDPN(v_i, j)$, for every i,j with $1 \leq i < j-1 \leq n-1$.

(5) α is an LMCS sequence iff for every i,j with $1 \leq i < j-1 \leq n-1$ there exists a $k \geq j$ depending upon i and j s.t. $LMPN_{v_k}(v_{j-1}, j) \geq LMPN_{v_k}(v_i, j)$.

Note that every MCS sequence is an LMCS sequence, because for every i,j, any k s.t. $v_k \in (N(v_{j-1}) \cap \{v_j, v_{j+1}, \ldots, v_n\})$ will satisfy Proposition 5.3.1(5).

Let Algorithm 1 = Algorithm MCS, Algorithm 2 = Algorithm LEX-BFS, Algorithm 3 = Algorithm MCBFS, Algorithm 4 = Algorithm MCDFS, and Algorithm 5 = Algorithm LMCS.

Let S_0 be the set of connected chordal graphs. Let $S_i = \{ G \in S_0$ s.t. every PEO of G can be generated by Algorithm i $\}$, $1 \leq i \leq 5$. Let $S(i,j) = \{ G \in S_0$ s.t. every PEO that can be generated by Algorithm i can also be generated by Algorithm j and conversely$\}$, $1 \leq i < j \leq 5$.

As mentioned in the introduction, it is reported [57] that S_1 and S_2 are proper subclasses of S_0. Below we show that S_i, $1 \leq i \leq 5$, is a proper subclass of S_0.

Proposition 5.3.2: S_i, $1 \leq i \leq 5$, is a proper subclass of S_0.

Proof: It is easy to see that the PEO $\alpha_1 = (1,2,3,4)$ of the graph G_1 in Figure 5.3.1 cannot be generated by Algorithm 5. So S_5 is a proper subclass

of S_0. Since All Algorithm i is a special case of Algorithm 5, $1 \le i \le 4$, and $i \ne 2$, S_i is a proper subclass of S_0, $1 \le i \le 4$, but $i \ne 2$. Again the PEO $\alpha_2=(1,2,3,4)$ of G_2 of Figure 5.3.1 cannot be generated by Algorithm 2. So S_2 is a proper subclass of S_0. ∎

Also it is reported [57] that S(1,2) is a proper subclass of S_0.

Let $G_{i,j}=G_3$ of Figure 5.3.1, $1 \le i < j \le 4$, but $(i,j) \ne (2,3)$. Let $G_{2,3}=G_4$, and $G_{2,5}=G_5$ of Figure 5.3.1. Let $\alpha_{1,2}=\alpha_{1,3}=\beta_{2,4}=(1,2,3,4,5)$, $\beta_{1,4}=\beta_{3,4}=(2,1,\ldots,5,4,3)$, $\alpha_{2,3}=(2,1,3,4,5,6)$, and $\alpha_{2,5}=(1,2,3,4,5,6,7)$. Let $\beta_{1,2}=\beta_{1,3}=\alpha_{2,4}=(2,1,3,4,5)$, $\alpha_{1,4}=\alpha_{3,4}=(1,2,5,4,3)$, $\beta_{2,3}=(1,2,3,4,5,6)$, and $\beta_{2,5}=(1,3,2,4,5,6,7)$. It is easy to verify that $\alpha_{i,j}$, and $\beta_{i,j}$ are PEOs of $G_{i,j}$ s.t. $\alpha_{i,j}$ can be generated by Algorithm i but not by Algorithm j, and $\beta_{i,j}$ can be generated by Algorithm j but not by Algorithm i, $1 \le i < j \le 4$, and $(i,j)=(2,5)$. Again S_5 is a proper subclass of S_0. Thus we have the following Proposition.

Proposition 5.3.3: (i) S(i,j) is a proper subclass of S_0, for $1 \le i,j \le 5$.

(ii) Algorithm i is different from Algorithm j, $1 \le i < j \le 5$.

In view of Propositions 5.3.2 and 5.3.3, it is natural to ask to characterize S_i, $1 \le i \le 5$, and S(i,j), $1 \le i,j \le 5$. In this section we characterize S_i, $1 \le i \le 5$, and S(i,j), $1 \le i,j \le 5$ except S(1,5).

The proofs of the following propositions are easy.

Proposition 5.3.4: If $(v_1,v_2,\ldots,v_n)$ is an MCS sequence of a k-tree G of order n, then

$MPN(v_i,j) \le k$ for all $i \le j < n-k+1$.

Proposition 5.3.5: If $(v_1,v_2,\ldots,v_n)$ is a PEO of a k-tree G, then $G[\{v_i,v_{i+1},\ldots,v_n\}]$ is a k-tree for all $i=1,2,\ldots,n-k+1$.

Now we characterize S_1.

Lemma 5.3.6: Let $G \in S_1$. Then $|S(G)| \ge 2$ and $\deg(s_i) = \deg(s_j)$ for all s_i, $s_j \in S(G)$.

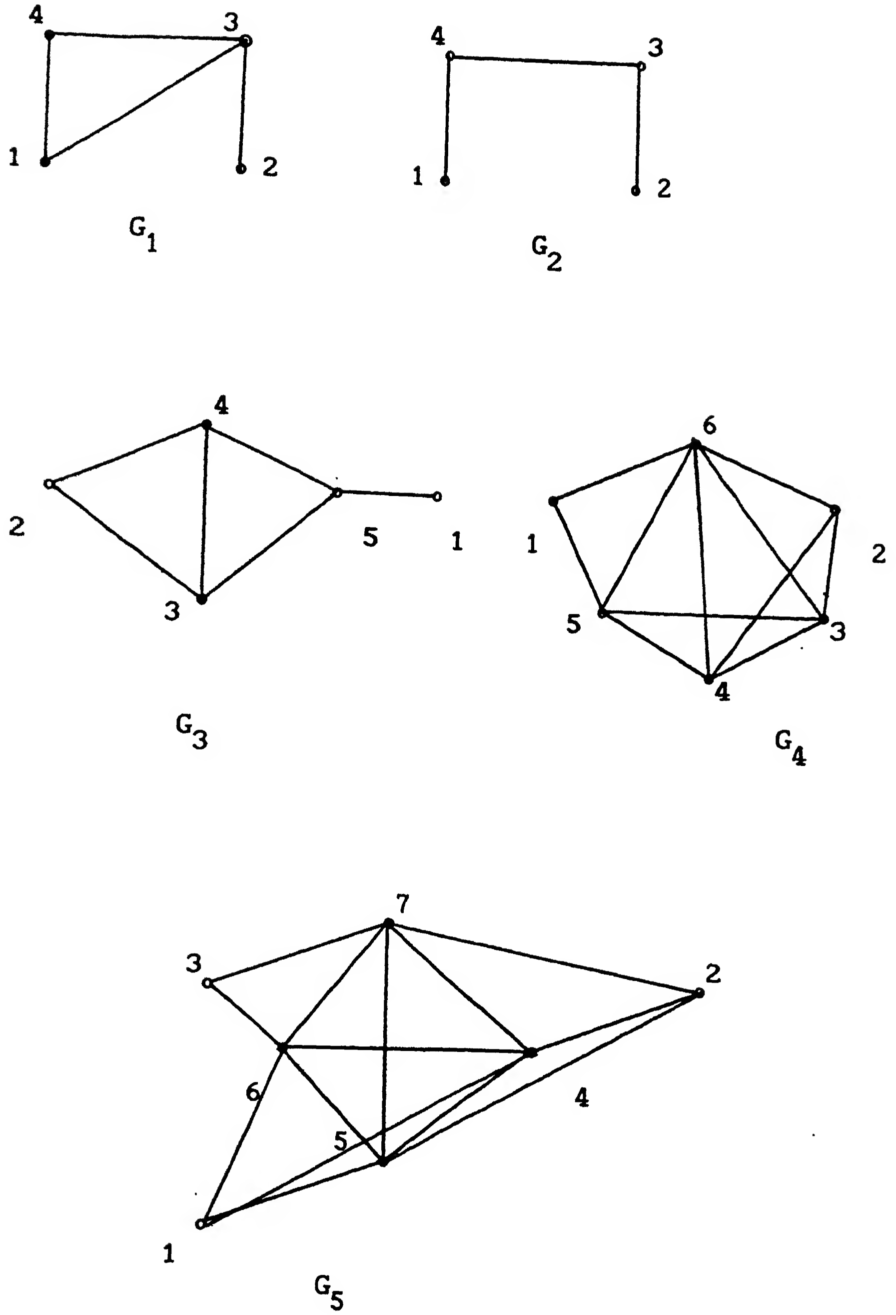


Figure 5.3.1: Examples Showing That $S_i \neq S_0$, $1 \le i \le 5$, and $S(i,j) \neq S_0$, $1 \le i < j \le 5$.

Proof: For a complete graph G our Lemma is easily seen to be true. So assume that G is not complete. That, $|S(G)| \geq 2$ follows from Theorem 1.4.2. If possible, let s_i, s_j be in S(G) and $deg(s_i) \neq deg(s_j)$, say $deg(s_i) > deg(s_j)$. Let $G' = G - \{s_i, s_j\}$. Then clearly G' is chordal. Let $\alpha = (\alpha(1), \alpha(2), \ldots, \alpha(n-2))$ be a PEO of G'. Then $\beta = (\beta(1), \beta(2), \ldots, \beta(n))$ is a PEO of G where $\beta(1) = s_i$, $\beta(2) = s_j$, and $\beta(i) = \alpha(i-2)$ for i, $3 \leq i \leq n$. But $MPN(\beta(2),3) = deg(s_j)$ and $MPN(\beta(1),3) = deg(s_i)$ if $s_i s_j \notin E$ and $MPN(\beta(2),3) = deg(s_j)-1$ and $MPN(\beta(1),3) = deg(s_i)-1$ if $s_i s_j \in E$. So in both cases $MPN(\beta(2),3) < MPN(\beta(1), 3)$. Hence by Proposition 5.3.1(1), $(\beta(1), \beta(2), \ldots, \beta(n))$ is not an MCS sequence of G. This contradicts the fact that $G \in S_2$. Hence $deg(s_i) = deg(s_j)$ for all s_i, s_j in S(G).∎

Lemma 5.3.7: If $G \in S_1$, then $G-v \in S_1$, for every $v \in S(G)$.

Proof: Assume that $G-v \notin S_1$, for some $v \in S(G)$. Let $\alpha' = (v_1, v_2, \ldots, v_{n-1})$ be a PEO of G-v that cannot be generated by MCS. Then clearly the PEO $\alpha = (v, v_1, v_2, \ldots v_{n-1})$ of G cannot be generated by MCS. So $G \notin S_1$, which is absurd.∎

Along the same line we can prove the following Lemma.

Lemma 5.3.8: If $G \in S_5$, then $G-v \in S_5$ for every $v \in S(G)$.

Theorem 5.3.9: $G \in S_1$ iff G is a k-tree for some $k \geq 1$.

Proof:

Sufficiency : Let G be a k-tree. Let $\alpha = (v_1, v_2, \ldots, v_n)$ be any PEO of G. We claim that α is an MCS sequence of G. First note that by Proposition 5.3.5 $G[\{v_{n-k+1}, v_{n-k+2}, \ldots, v_n\}]$ is a complete graph on k vertices. So $(v_{n-k+1}, v_{n-k+2}, \ldots, v_n)$ can be chosen by MCS algorithm as an end partial initial sequence. we apply induction on i, where $n-k+1 \geq i \geq 1$ to establish our claim. For $i = n-k+1$, our claim is true. Let us assume that $(v_i, v_{i+1}, \ldots, v_n)$, $1 < i < n-k+1$, has been chosen by MCS. Now by Proposition 5.3.5, $G[\{v_{i-1}, v_i, \ldots, v_n\}]$ is a k-tree. so $MPN(v_{i-1}, i) = k$. Again by

proposition 5.3.4, $MPN(v_j, i) \le k$ for any MCS sequence of a k-tree for every $i<n-k+1$ and $j < i$. So v_{i-1} can be chosen by MCS algorithm as the next vertex, i.e, $(v_{i-1}, v_i, \ldots, v_n)$ can be chosen by MCS Algorithm as the end partial initial sequence. Hence, by induction principle, α is an MCS sequence. Therefore $G \in S_1$.

Necessity:

Let $G \in S_1$. Let $v \in S(G)$, and let $deg(v) = k$ (≥ 1). We claim that G is a k-tree. We prove this by induction on n, the order of G. If G is complete or n = 2, 3 it is easy to check that G is a k-tree. Assume that our claim is true for all $G \in S_1$ having fewer than n (>3) vertices. Let $G \in S_1$ and G has n vertices. Then by Lemma 5.3.6, and by Theorem 1.4.2 there exist two non adjacent simplicial vertices u and w of G s.t. deg(u) = deg(w) = deg(v) = k. Let G′= G-u. Now as u and w are non adjacent vertices of G, the degree of w in G′ is k. Also by Lemma 5.3.7, $G' \in S_1$. So by induction hypothesis, G′ is a k-tree. Since u is a simplicial vertex of G and deg(u) = k, G is a k-tree .■

Theorem 5.3.10: $G \in S_5$ iff G is a k-tree for some $k \ge 1$.

Proof: The sufficiency follows from Theorem 5.3.9 as $S_1 \subseteq S_5$, because MCS is a special case of LMCS.

Necessity:

Let $G \in S_5$. Let $v \in S(G)$, and deg(v)=k. We claim that G is a k-tree. We induct on n, the order of G. For n=2,3, and 4 it is easy to check that our claim is true. Assume that our claim is true for all $G \in S_5$ having less than n vertices. Let $G \in S_5$ be of order n. If G is complete it is a k-tree. Assume that G is non-complete. So by Theorem 1.4.2, G has two non-adjacent simplicial vertices, say x and y. Let $deg(x) = k_1$ and $deg(y)=k_2$. By Lemma 5.3.8, $G-x \in S_5$ and $G-y \in S_5$. So by induction principle, G-x is a k_2-tree and G-y is a k_1-tree. If G-{x,y} is not complete, then $k_1=k_2$, and G is a

k-tree. If $G-\{x,y\}$ is not complete, then either $N(x) \subseteq N(y)$ or $N(y) \subseteq N(x)$. In this case, if possible let $k_1 \neq k_2$. Wlg, $k_1 < k_2$. So $N(x) \subset N(y)$. Let $\alpha=(v_1,v_2,\ldots,v_{n-2})$ be a PEO of $G-\{x,y\}$. Then $\alpha' = (y,x,v_1,v_2,\ldots,v_{n-2})$ is a PEO of G that cannot be generated by LMCS, which is a contradiction to the fact that $G \in S_5$. So $k_1=k_2=k$, and G is a k-tree.∎

Lemma 5.3.11: Let $S=\{z_1,z_2,\ldots,z_k\}$ be a minimum cardinality cut set of G. Let $G_i=(V_i,E_i)$, $1 \le i \le r$, $r \ge 2$ be the connected components of G-S. If $G \in S_j$, $2 \le j \le 4$, then $|V_i| = 1$ for all i, $1 \le i \le r$.

Proof: We prove the Lemma for $G \in S_2$. For $G \in S_3$, and $G \in S_4$, the Lemma can be proved in similar lines.

Assume that $G \in S_2$. If possible, suppose some G_i has more than one vertex. Wlg, we may assume that G_1 has more than one vertex. So we can find a PEO $\alpha = (v_{11}, v_{12},\ldots,v_{1m_1},\ldots,v_{r1}, v_{r2},\ldots,v_{rm_r}, z_1, z_2,\ldots,z_k)$ of G where $v_{ij} \in V_i$, $1 \le i \le r$, $1 \le j \le m_i$, and $|V_i|=m_i$. Now $(v_{11}, v_{12},\ldots, v_{1m_1-1}, v_{21},v_{22},\ldots,v_{2m_2},\ldots,v_{r1},\ldots, v_{rm_r}, z_1, z_2,\ldots,z_k, v_{1m_1})$ is a PEO of G. Since $v_{1m_1} \in G_1$, there exists a vertex v_{1j}, $1 \le j \le m_1-1$, s.t. $v_{1m_1}v_{1j} \in E(G_1)$. Let $(v_{11},v_{12},\ldots,v_{1m_1-1},v_{21},v_{22},\ldots,v_{2m_2},\ldots,v_{r1},\ldots, v_{rm_r},z_1,z_2,\ldots, z_k, v_{1m_1})=(v_1,v_2,\ldots,v_n)$. So $v_{n-k-1} = v_{rm_r}$. Let $v_{1j} = v_i$. But $\mathrm{LEXPN}(v_{n-k-1}, n-k)^{-1} <_{lex} \mathrm{LEXPN}(v_i, n-k)$, as $r \ge 2$. So by Proposition 5.3.1(2), the above sequence is not a LEX-BFS sequence of G, contradicting the fact that $G \in S_2$. Thus $|V_i|=1$, $1 \le i \le r$.∎

Next we characterize S_i, $2 \le i \le 4$. To this end we need the following definition.

A k-tree G is called a p-k-tree if $|S(G)|=n-k$.

Theorem 5.3.12: $G \in S_j$, $2 \le j \le 4$ iff G is a P-k-tree for some $k \ge 1$.

Proof: We prove the theorem for j=2. The proofs for the cases j=3 and j=4 go in the same lines.

Necessity :

Let $G \in S_2$. let S be a minimum cardinality cutset of G and $|S| = k$. We claim that G is a p-k-tree. Now by the minimality of S, G[S] is a complete graph on k vertices. Let G_i, $1 \le i \le r$, $r \ge 2$ be the connected components of G-S. By Lemma 5.3.11, each G_i has exactly one vertex. Since S is a minimum cardinality cutset of G $\deg(v) \ge k$ for all $v \in V$. So G is a p-k-tree.

Sufficiency :

Let G be a p-k-tree of order n with $S = \{z_1, z_2, \ldots, z_k\}$, the set of all vertices of degree n-1. Let $V\text{-}S = \{v_1, v_2, \ldots, v_{n-k}\}$. Then any PEO of G is of the following type:

$(v_{\pi(1)}, v_{\pi(2)}, \ldots, v_{\pi(n-k-1)}, y_1, y_2, \ldots, y_{k+1})$ where $y_1, y_2, \ldots, y_{k+1}$ is any permutation on $\{z_1, z_2, \ldots, z_k, v_{\pi(n-k)}\}$ and π is a permutation on $\{1, 2, \ldots, n-k\}$. It can be easily seen that the above sequence is a LEX-BFS sequence of G. This completes the proof of the sufficiency.∎

Lemma 5.3.13: Let G be a connected chordal graph having no induced path of length 3. Let S be a minimum cardinality cutset of G. Let $G_i(V_i, E_i)$, $1 \le i \le r$, $r \ge 2$ be the connected components of G-S. Then each $v \in V_i$, $1 \le i \le r$, $r \ge 2$ is adjacent to every vertex of S.

Proof: We claim that each vertex of G_i, $1 \le i \le r$, $r \ge 2$ is adjacent to every vertex of S. If possible, there exists some G_i s.t. $v \in V_i$ and $vz \notin E(G)$ for some $z \in S$. Since each vertex of S is adjacent to some vertex of G_i, $1 \le i \le r$, $G' = G[V_i \cup \{z\}]$ is connected. Let P be a shortest path from v to z in G'. Let $w \in G_j$, $i \ne j$ be s.t. $zw \in E(G)$. Existence of such a vertex is assured by the fact that $r \ge 2$ and by the minimality of S. Now $P \cup \{zw\}$ is an induced path of length at least 3. This contradicts the fact that G has no induced path of length 3. Thus our claim is established .∎

Theorem 5.3.14: $G \in S(1,2)$ iff G is a chordal graph and G does not contain any P_4 as an induced subgraph.

Proof: Necessity:

If possible, suppose x, x_1, y_1, y be an induced path of length 3. Let $\alpha=(v_1, v_2,\ldots,v_n)$ be an MCS sequence of G s.t. $v_n = y_1$ and $v_{n-1} = x_1$. Note that such an MCS sequence can always be constructed. Let $x=v_i$, and $y=v_j$. If $i > j$, then α is not a LEX-BFS sequence of G as $xy_1 \notin E(G)$, but $yy_1 \in E(G)$. So assume that $i < j$. Now construct an MCS sequence $\alpha'=(v'_1, v'_2,\ldots, v'_n)$, where $v'_n = v_{n-1}$, $v'_{n-1} = v_n$, and $v'_i = v_i$, $i \neq n,n-1$. Then clearly α' is an MCS sequence but not a LEX-BFS sequence as above. This contradicts the fact that $G \in S(1,2)$. Thus G has no induced P_4.

Sufficiency :

Let G be a connected chordal graph with n vertices having no induced path of length 3. We claim that $G \in S(1,2)$. We prove this by induction on the order of G. If G is complete or n=2,3, our claim is easily seen to be true. Let G be a connected chordal graph of order n having no induced path of length 3. Let G_i, $1 \le i \le r$, $r \ge 2$, be the connected components of G-S, where $S =\{z_1,z_2,\ldots,z_k\}$ is a minimum cardinality cutset of G. Then by Lemma 5.3.13 and by the minimality of S, $\deg(s) = n-1$ for all $s \in S$ and $\deg(v) < n-1$ for all $v \in V-S$. Let $(v_1, v_2,\ldots, v_n)$ be an MCS sequence of G. Then the last k+1 vertices of the above sequence contains a vertex of G_j, for some j , $1 \le j \le r$, say of G_1. Then $\alpha'=(v_{n-k-m_1+1},\ldots, v_{n-k+1},\ldots, v_n)$ is an MCS sequence of $H_1=G[V_1 \cup S]$, where m_1 is the number of vertices of G_1. Let $v_{\pi(1)}, v_{\pi(2)},\ldots, v_{\pi(k)}$ be the order of the vertices of S in which they appear in α' s.t. $\pi(i) < \pi(j)$ if $i < j$. Then $(v_1, v_2,\ldots, v_{n-k-m_1}, v_{\pi(1)},\ldots,v_{\pi(k)})$ is an MCS sequence of $G'= G[V-V_1]$. As G' has no induced path of length 3, by the induction hypothesis, the above sequence is a LEX-BFS sequence of G'. Also by the induction hypothesis, $(v_{n-k-m_1+1},$

$\ldots, v_n)$ is a LEX-BFS sequence of H_1. It can be easily seen from the structure of G in terms of G_i, $1 \le i \le r$, that $(v_{n-k-m_1+1}, \ldots, v_n)$ can be chosen as the final partial LEX-BFS sequence of G, i.e. there exists a LEX-BFS sequence β of G s.t. $(v_{n-k-m_1+1}, \ldots, v_n)$ is the last m_1+k vertices of β. Since LEX-BFS priority of the vertices of $S' = \{v_{n-k-m_1}, v_{n-k-m_1-1}, v_1\}$ depend only on $z_1, z_2, \ldots, z_k$ and on S', $(v_1, v_2, \ldots, v_n)$ is a LEX-BFS sequence of G as $z_{\pi(1)}, z_{\pi(2)}, \ldots, z_{\pi(k)}$ appear in the same order as they appear in $(v_{n-k-m_1+1}, \ldots, v_n)$. So the above sequence is a LEX-BFS sequence. By a similar argument it can be shown that every LEX-BFS sequence of G is again an MCS sequence of G. This completes the proof of the sufficiency.∎

Let $\alpha=(v_1, v_2, \ldots, v_n)$ be any MCBFS sequence of G. For every $v_i \in V$, let $P_\alpha(v_i)=v_j$ if v_i is numbered while v_j is in the front of the queue in the MCBFS Algorithm. We take $P_\alpha(v_n)= v_n$. Note that $P_\alpha(v_i)= v_j$ iff j is the largest index s.t. $v_iv_j \in E(G)$. Define a rooted tree $T(\alpha)$ of G w.r.t. α as follows: $V(T(\alpha))= V(G)$, and $E(T(\alpha))= \{ v_iP_\alpha(v_i),\ 1 \le i \le n-1 \}$, and $\mathrm{root}(T(\alpha))= v_n$. Define $L(v_n)= 0$, and $L(v_i)= L(P_\alpha(v_i)) + 1$, $1 \le i \le n-1$.

Before characterizing S(2,3), we first present some observations on $T(\alpha)$.

Observation 5.3.15: Let v_i and v_j be in $V(T(\alpha))$ s.t. $L(v_i)= L(v_j)$. If $v_iv_j \in E(G)$ and $i > j$, then $v_iP_\alpha(v_j) \in E(G)$.

Proof: Since $i > j$, either $P_\alpha(v_i) = P_\alpha(v_j)$ or $\alpha(P_\alpha(v_i)) > \alpha(P_\alpha(V_j))$. If $P_\alpha(v_i)=P_\alpha(v_j)$, then clearly $v_iP_\alpha(v_j) \in E(G)$. Assume that $P_\alpha(v_i) \ne P_\alpha(v_j)$. Let $Q_1(v_n, v_i)$ and $Q_2(v_n, v_j)$ be the paths from v_n to v_i and v_n to v_j in $T(\alpha)$, respectively. Then $C= Q_1(v_n,v_i) \cup Q_2(v_n,v_j) \cup \{ v_iv_j\}$ will contain a chord less cycle of length at least four if $v_iP_\alpha(v_j) \notin E(G)$. This contradicts the chordality of G.∎

Since $T(\alpha)$ is a BFS tree, we have the following:

Observation 5.3.16: For every $v_iv_j \in E(G)$, $|\, L(v_i) - L(v_j)| \le 1$.

The following result characterizes $S(2,3)$.

Theorem 5.3.17: $G \in S(2,3)$ iff G is a chordal graph and G does not contain any $P_4 + v$ (P_4+v is obtained by taking a vertex v and joining it to every vertex of a P_4) as an induced subgraph.

Proof: Necessity:

Assume that G contains a P_4+v as an induced subgraph. Let x, x_1, y_1, y be the P_4 in the P_4+v. Let $\alpha = v_1, v_2, \ldots, y_1, x_1, v$ be an MCBFS sequence of G. Such a sequence can be constructed as $\{y_1, x_1, v\}$ induces a complete subgraph of G. Let $x = v_i$, and $y = v_j$. If $i > j$, then clearly α cannot be generated by LEX-BFS as $yy_1 \in E(G)$ but $y_1x \notin E(G)$. So assume that $i < j$. Let $\alpha' = v_i, v_{i+1}, \ldots, v_j, x_1, y_1, v$. Since $v_iv \in E(G)$, and α' is obtained from a subsequence of α by interchanging the positions of x_1 and y_1, we can get a MCBFS sequence β s.t. $\beta(k) = \alpha'(k)$, $i \le k \le n$. Now clearly β cannot be generated by LEX-BFS algorithm.

Sufficiency:

Let G be a P_4+v free chordal graph having minimum number of vertices s.t. $G \notin S(2,3)$.

Case 1: There exists an MCBFS sequence, say $\alpha = (v_1, v_2, \ldots, v_n)$ which is not a LEX-BFS sequence.

Let i be the largest index s.t. v_i cannot be chosen by LEX-BFS. By the choice of G, i=2. Now $MCBPN(v_2,3) \ge MCBPN(v_1,3)$ but $LEXPN(v_2,3) < LEXPN(v_1,3)$. Let $P_\alpha(v_2) = v_i$. Since α is not a LEX-BFS sequence we can choose v_j s.t. j is the largest index s.t. $v_1v_j \in E(G)$ but $v_2v_j \notin E(G)$. Again $MCBPN(v_2,3) \ge MCBPN(v_1,3)$. So there exists v_k with largest index k s.t. $v_kv_2 \in E(G)$ but $v_kv_1 \notin E(G)$, since $P_\alpha(v_2) = v_1$, $i > j > k$. Again v_j can be chosen so that $v_sv_1 \in E(G)$, and $s > j$ implies $v_sv_2 \in E(G)$. Note that

$P_\alpha(v_1)=v_i$ as α is an MCBFS sequence. Since v_1 and v_2 are simplicial vertices of G and $G-v_1$, respectively, $v_iv_j \in E(G)$ and $v_iv_k \in E(G)$. Now v_jv_k and v_2v_1 cannot simultaneously belong to E(G), otherwise v_1, v_2, v_j, v_k will form a chordless 4-cycle. Again if either $v_1v_2 \in E(G)$ or $v_jv_k \in E(G)$, then $G[\{v_i, v_j, v_k, v_1, v_2\}]$ will be isomorphic to a P_4+v. So neither $v_1v_2 \in E(G)$ nor $v_jv_k \in E(G)$.

Subcase 1(a): $L(v_i) = L(v_j) = L(v_k)$.

Let $P_\alpha(v_j)=v'_j$ and $P_\alpha(v_k)= v'_k$. Now by observation 1, $v_iv'_j \in E(G)$, and $v_iv'_k \in E(G)$. If $v'_j=v'_k$, then $G[\{v_i, v'_j, v_j, v_k, v_1\}]$ will be isomorphic to P_4+v. So assume that $v'_j \neq v'_k$. If $v_jv'_k \in E(G)$, then $G[\{v_1, v_k, v'_k, v_j, v_i\}]$ will be isomorphic to a P_4+v. So assume that $v_jv'_k \notin E(G)$. Now $v'_jv'_k \in E(G)$, otherwise $Q_1(v'_j, v_n) \cup Q_2(v'_k, v_n) \cup \{ v_iv'_j\} \cup \{v_iv'_k\}$ will contain an induced chordless cycle of length at least 4, where $Q_1(v'_j, v_n)$, and $Q_2(v'_k, v_n)$ are paths in $T(\alpha)$ from v_n to v'_j and v'_k, respectively. Then $G[\{v'_j, v'_k, v_j, v_k, v_i\}]$ will be isomorphic to a P_4+v.

Subcase 1(b): $L(v_i) = L(v_j)$ and $L(v_j)= L(v_k)+1$.

Suppose $P_\alpha(v_k)=v_i$. By the choice of G, there exists s s.t. $v_sv_k \in E(G)$ but $v_sv_2 \in E$, and $i > s > j$. Now $v_iv_s \in E(G)$ as v_k is a simplicial vertex of $G[\{v_k, v_{k+1}, \ldots, v_n\}]$. Let $P_\alpha(v_s) = v'_s$. Then $v'_sv_i \in E(G)$. By the choice of v_j, $v_1v_s \notin E(G)$. So $G[\{v'_s, v_s, v_k, v_2, v_i\}]$ is isomorphic to a P_4+v. Assume that $P_\alpha(v_k) = v'_k \neq v_i$. Then $\alpha(v'_k) > \alpha(v_i)$ and $L(v'_k) = L(v_i)$. Let $P_\alpha(v_i) = v_i''$. Then clearly $v_i''v'_k \in E(G)$ and $v_iv'_k$. So $G[\{ v_i'', v'_k, v_k, v_1, v_i\}]$ is isomorphic to a P_4+v.

Subcase 1(c): $L(v_j) = L(v_k)$ and $L(v_i) = L(v_j)+1$.

If $P_\alpha(v_j) = v'_j$ and $v'_j\neq v_i$, then $G[\{P_\alpha(v_i), v'_j, v_j, v_1, v_i\}]$ is isomorphic to a P_4+v, as $P_\alpha(v_i)v'_j \in E(G)$, and $v_iv'_j$. So assume that $P_\alpha(v_j) = v_i$. So $P_\alpha(v_k)=v_i$. So there exists v_s s.t. $v_sv_k \in E(G)$, $v_sv_1 \notin E(G)$, and $i > s > j$. If $L(v_s) = L(v_j)$, Then wlg, we may assume that $P_\alpha(v_s) \neq v_i$(otherwise we

will consider v_s, v_j, v_k, v_1). Then $G[\{P_\alpha(v_s), P_\alpha(v_1), v_s, v_1, v_k\}]$ is isomorphic to a P_4+v. So $L(v_s) = L(v_j)+1$. Then $G[\{P_\alpha(v_s), v_1, v_s, v_k, v_2\}]$ is isomorphic to a P_4+v, since $v_2v_s \notin E(G)$, as $v_1v_s \notin E(G)$.

Case 2: There exists a LEX-BFS sequence $\alpha = (v_1, v_2, \ldots, v_n)$ s.t. α is not an MCBFS sequence.

Let i be the smallest index s.t. $\alpha' = (v_i, v_{i+1}, \ldots, v_n)$ is an MCBFS sequence of $G[\{v_i, v_{i+1}, \ldots, v_n\}]$. By the choice of G, i=3.

So MCBPN(v_2,3) < MCBPN(v_1,3) - - - - ---------(1).

But LEXPN(v_2,3) ≥ LEXPN(v_1,3) -------------------(2).

So by (1) LEXPN(v_2,3) > LEXPN(v_1,3). Now consider $\beta = (v_2, v_1, v_3, \ldots, v_n)$. Now β is an MCBFS sequence which is not a LEX-BFS sequence. So by Case 1, G will contain a P_4+v as an induced subgraph contrary to our assumption. Hence the result. ∎

Theorem 5.3.18: $G \in S(3,4)$ iff G is a chordal graph and G does not contain any P_4 as an induced subgraph.

Proof: Necessity: Assume that P= x, x_1, y_1, y be an induced P_4 of G. Let $\alpha = (v_1, v_2, \ldots, v_{n-2}, x_1, y_1)$ be an MCDFS sequence of G. Let $x=v_i$, and $y=v_j$. If $i > j$, then α cannot be generated by MCBFS as $xy_1 \notin E(G)$ but $y_1y \in E(G)$. So assume that $i < j$. If $v_sy_1 \notin E(G)$ for some s,$n-2 \le s \le j$, then α cannot be generated by MCBFS, because $v_j \in N(y_1)$ but $v_1 \notin N(y_1)$, and $j < s$. Again $v_{n-2}x_1 \in E(G)$ and $v_{n-2}y_1 \in E(G)$. So we can construct an MCDFS sequence β s.t. $\beta(k) = \alpha(k)$ for all k, $i \le k \le n-2$, $\beta(n-1) = y_1$ and $\beta(n) = x_1$. Now as above β cannot be generated by MCBFS Algorithm.

Sufficiency:

The sufficiency follows in the same line as in the sufficiency of Theorem 5.3.14. ∎

Theorem 5.3.19: $G \in S(1,3)$ iff G is a chordal graph and G does not contain any P_4 as an induced subgraph.

Proof: Necessity: Assume that $P=x,x_1,y_1,y$ be an induced P_4 of G. Let $\alpha=(v_1,v_2,\ldots,v_{n-2},x_1,y_1)$ be an MCS sequence of G. Let $x=v_i$, and $y=v_j$. If $i > j$, then α cannot be generated by MCBFS as $xy_1 \notin E(G)$, but $yy_1 \in E(G)$. So assume that $i < j$. Then $\alpha'=(v_1,v_2,\ldots,v_{n-2},y_1,x_1)$ can be generated by MCS but cannot be generated by MCBFS as above.

Sufficiency:

This follows in the same lines of Theorem 5.3.14.∎

Proposition 5.3.20: If $G \in S(1,4)$ then $G-v \in S(1,4)$ for every $v \in S(G)$.

Proof: If not, let $G-v \notin S(1,4)$. Suppose there exists an MCDFS sequence $\alpha'=(v_1,v_2,\ldots,v_{n-1})$ which is not an MCS sequence. Let $\min\{\alpha'(x)$ s.t. $x \in N(v)\}=k$. Now construct an MCDFS sequence α from α' by suitably incorporating v s.t. $\alpha(v) < k$. This is possible since $N(v) \subseteq N(x)$ for all $x \in N(v)$. Now clearly α is not an MCS sequence since $\alpha(x) < k$. So $G \notin S(1,4)$, which is a contradiction to our assumption. If there exists an MCS sequence of G-v which is not an MCDFS sequence, then using a similar method as above, we can show that $G \notin S(1,4)$.∎

Theorem 5.3.21: $G \in S(1,4)$ iff G is chordal and G does not have any P_4 as an induced subgraph.

Proof: Necessity: If not, Let G have a P_4 as an induced subgraph. Let $S=\{z_1,z_2,\ldots,z_k\}$ be a minimum cardinality cut set of G. Let $H_i=(V_i,E_i)$, $1 \le i \le r$, $r \ge 2$ be the connected components of G-S. Let $G_i=G[V_i \cup S]$, $1 \le i \le r$. In view of Proposition 5.3.20, wlg, we may assume that $r=2$, $|V_2|=1$, $|V_1|=2$, and V_1 contains a vertex, say v_2 s.t. there exists a vertex of S, say z_i s.t. $v_2z_i \notin E(G_1)$. The existence of v_2 is assured since G has an induced P_4. Since $|S|=k$, $\deg(v) \ge k$ for all $v \in V(G)$. Let $V_2=\{v_3\}$ and $V_1=\{v_1,v_2\}$. Now the MCS sequence α, where $\alpha=(v_3,v_2,z_i, z_1,z_2,\ldots,z_{i-1},z_{i+1},\ldots z_k,v_1)$ cannot be generated by MCDFS Algorithm as $z_iv_2 \in E(G)$. So G has no induced P_4.

Sufficiency:

The line of proof is that of Theorem 5.3.14.∎

Theorem 5.3.22: G $\in$ S(4,5) iff G is chordal and G does not contain any P_4 as an induced subgraph.

Proof: Necessity follows from the fact that MCS is a special case of LMCS, and sufficiency goes in the same lines as that of Theorem 5.3.14.∎

Theorem 5.3.23: G $\in$ S(3,5) iff G is chordal and G does not contain P_4 as an induced subgraph.

Proof: Necessity: This follows from the fact that G $\in$ S(1,3) implies G is P_4 free and every MCS sequence is an LMCS sequence.

Sufficiency: Let G be a P_4 free chordal graph. Then G $\in$ S(3,4) and G $\in$ S(4,5), whence G $\in$ S(3,5).∎

Theorem 5.3.24: G $\in$ S(2,5) iff G is chordal and G has no P_4 as an induced subgraph.

Proof: Necessity: Same argument as in Theorem 5.3.14.

Sufficiency: Let G be a chordal graph free from P_4. Then G $\in$ S(3,5) and G $\in$ S(2,3). So G $\in$ S(2,5).∎

Theorem 5.3.25: G $\in$ S(2,4) iff G is chordal and G contains no P_4 as an induced subgraph.

Proof: Necessity: Same argument as in Theorem 5.3.14.

Sufficiency: Let G be a chordal graph s.t. G contains no P_4 as an induced subgraph. Then G $\in$ S(2,3) and G $\in$ S(3,4). So G $\in$ S(2,4).∎

We are unable to characterize S(1,5). However, below we show that P_4-free chordal graphs are a subclass of S(1,5).

Theorem 5.3.26: If G is a chordal graph and G does not contain any P_4 as an induced subgraph, then G $\in$ S(1,5).

Proof: Since G is a chordal graph and G does not contain any P_4 as an induced subgraph, G $\in$ S(1,2), and G $\in$ S(2,5). Hence G $\in$ S(1,5).∎

We are unable to chararacterize S(1,5) and left this an an open

problem.

Now we indicate linear time recognition algorithms for the classes S(i), $1 \le i \le 5$. The class of k-trees can be recognized in O(n+m) time. For example, the following Lemma, whose proof is easy, gives a linear time recognition algorithm of k-trees.

Lemma 5.3.27: Let G be a connected chordal graph and let $(v_1, v_2, \ldots, v_n)$ be a PEO of G. Then G is a k-tree iff $\deg_{G^i}(v_i)=k$ where $G^i = G[\{ v_i, v_{i+1}, \ldots, v_n\}]$, $1 \le i \le n-k+1$.

Now we have the following result .

Theorem 5.3.28: The class S_1 and S_2 can be recognized in O(n+m) time.

The following lemma whose proof follows from the definition of p-k-trees, gives a characterization of p-k-trees.

Lemma 5.3.29: Let G be a connected graph of order n. Let $S = \{v \in V$ s.t. $\deg(v)=n-1 \}$. Then G is a p-k-tree with $k \ge 1$ iff $| S | = k$ and for all $v \in V - S$, $\deg(v) = k$.

Since the condition of the above lemma can be tested in O(n+m) time, we have the following result.

Theorem 5.3.30: The classes S_2, S_3 and S_4 can be recognized in O(n+m) time.

Corneil *et al* [30] have suggested a linear time algorithm to recognize the graphs having no induced path of length three (CO graphs see[37]). Using this fact and the fact that connectedness and chordality of a graph can be tested in O(n+m) time, we have the following result.

Theorem 5.3.31: The class S(i,j), $1 \le i,j \le 5$, except S(1,5), and S(2,3) can be recognized in O(n+m) time.

Since P_4-free chordal graph can be recognized in linear time, we can recognize P_4+v-free chordal graphs in O(nm) time. So we have the following result:

Theorem 5.3.32: The class S(2,3) can be recognized in O(nm) time.

5.4 Hamiltonian Elimination Orderings And Jump Number:

Let $\alpha = (v_1, v_2,\dots,v_n)$ be a PEO of a chordal graph G. The jump number $J(\alpha)$ is the number of consecutive pairs of vertices in α which are nonadjacent in the graph G. The jump number J(G) of a chordal graph G is min $\{ J(\alpha) : \alpha$ is a PEO of G $\}$. A PEO α for which $J(\alpha) = J(G)$ is called a jump sequence of G.

Recall that S(G) is the set of all simplicial vertices of a chordal graph G. Note that in a non-complete k-tree G, no two vertices of S(G) are adjacent and each vertex of S(G) has degree k. Now define a relation R on S(G) by u R v, $u,v \in V$, iff either $u = v$ or $uv \in E$. Clearly the relation R is an equivalence relation on S(G). For $v \in S(G)$, denote [v] by the equivalence class containing v. The number of equivalence classes of S(G) under R is denoted by e(G).

Jamison and Laskar [76] proved that $J(G) \le e(G) - 2$ for a chordal graph G and showed that there are chordal graphs for which $e(G)- J(G)$ is arbitrarily large. However, they proved in [76] that for a tree T, $J(T) = e(T)-2$.

In this section we present two characterizations of H-perfect k-trees. We prove that for a k-tree, $J(G) = e(G) -2$, which extends the result for 1-trees due to Jamison and Laskar [76]. We also present a linear time algorithm to find the jump number and a jump sequence of k-trees. We then present a polynomial recognition algorithm for H-perfect chordal graphs.

To this end we first introduce some new concepts and prove some results on k-trees.

A k-tree G of order n is called a p-k-tree if G has exactly k vertices of degree n-1. A graph G of order n is called a nearly p-k-tree if it has a (k+1)-clique C and the remaining (n-k-1) vertices are joined to exactly k vertices of C and only to these k vertices of G s.t. there exist three vertices of C and only to these k vertices of G s.t. there exist three

vertices, say v_1, v_2, and v_3 outside C with $N(v_i) \neq N(v_j)$, $1 \le i < j \le 3$.

Lemma 5.4.1: [76]: For a chordal graph G, $e(G) \ge e(G-v)$, for every $v \in S(G)$.

Lemma 5.4.2: Let G be a k-tree and H be an induced sub k-tree of G. If $S(G) \subseteq V(H)$, then $H = G$.

Proof : The proof is by induction on n, the order of G. If G is either complete or n=k or k+1 our lemma is easily seen to be true. Let G be a k-tree with n ($n \ge k+2$) vertices. Let u be any simplicial vertex of G. Now $d(u) = k$. As $S(G) \subseteq V(H)$ and H is a k-tree, u is a simplicial vertex of H and $N_G(u) \subseteq V(H)$. Hence $S(G-u) \subseteq V(H-u)$. Therefore by the induction principle, $H-u = G-u$, whence $H = G$.∎

Let G be a k-tree of order n and H be an induced sub k-tree of order n_1 of G. The graph G is said to be constructible from H if there exists a PEO $\alpha = (v_1, v_2, \ldots, v_n)$ of G s.t. the end partial sequence ($v_{n-n_1+1}, \ldots, v_n$) of length n_1 is a PEO of H. In this case α is said to be a constructing PEO of G w.r.t. H.

Remark 5.4.3: If G is constructible from H, then given any PEO $\beta = (w_1, w_2, \ldots, w_{n_1})$ of H, one can find a constructing PEO $\alpha = (v_1, v_2, \ldots, v_n)$ of G s.t. $v_{n-n_1+i} = w_i$, $1 \le i \le n_1$.

Lemma 5.4.4: Let G be a k-tree and H be any induced sub k-tree of G. Then G is constructible from H.

Proof: For n=k or k+1 it can be easily checked that our lemma is true. Let G be a k-tree on n vertices ($n \ge k+2$) and H be any induced sub k-tree of G. If $S(G) \subseteq V(H)$, then by lemma 5.4.2, $G = H$. So G is constructible from H. Suppose S(G) is not a subset of V(H). Let $v \in S(G)-V(H)$. Then G-v is a k-tree and H is an induced sub k-tree of G. So by induction principle, G-v is constructible from H. Let $\alpha = (v_1, v_2, \ldots, v_{n-1})$ be a constructing PEO of G-v w.r.t. H. Then $\beta = (v, v_1, v_2, \ldots, v_{n-1})$ is a constructing PEO of G

w.r.t. H. Therefore G is constructible from H.∎

The following result of Proskurowski [14] is a special case of Lemma 5.4.4.

Corollary 5.4.5[103]: Any k-clique of a k-tree G can be made a basis of G.

Let $\alpha = (v_1, v_2, \dots, v_n)$ be any PEO of a chordal graph G. We denote $(v_1, v_2, \dots, v_i)$ by $\alpha(i)$, and v_i by $\alpha[i]$.

Let G be a chordal graph and v be any simplicial vertex of G. Let $\alpha =(v_1, v_2, \dots, v_n)$ be any PEO of G. Define $L_G(\alpha) = s$ if there exists $s \le n-1$ s.t. $v_i v_{i+1} \in E$ for all $i, 1 \le i \le s-1$ but $v_s v_{s+1} \notin E$. Otherwise $L_G(\alpha) = n$. We define $L_G(v) = \text{Max} \{ L_G(\alpha) \text{ s.t. } \alpha[1] = v\}$.

Note that if α is an HEO of G, then $L_G(\alpha)=n$, and if further $\alpha[1] = v$, then $L_G(v) = n$. We call a chordal graph G with $e(G) \ge 3$ an extremal chordal graph if $J(G) = e(G) -2$.

The following proposition follows from the fact that $J(G) \le e(G)-2$ for a chordal graph G[76].

Proposition 5.4.6: Let G be a chordal graph. If $e(G) = 2$, then G is H-perfect. Moreover, any simplicial vertex can be made the starting vertex of an HEO of G.

Lemma 5.4.7: If G is an extremal chordal graph, then any simplicial vertex can be made the starting vertex of some jump sequence of G.

Proof : Let G be a chordal graph of order n and $e(G) = m \ge 3$. Let $v_1 \in S(G)$ and $L_G(v_1) = m_1$. Let α be a PEO of G s.t. $L_G(\alpha) = m_1$ and $\alpha[1] = v_1$. Let $\alpha(m_1) = (v_1, v_2, \dots, v_{m_1})$ and $G' = G - \{ v_1, v_2, \dots, v_{m_1} \}$. Then $e(G')= e(G)-1$. If $m > 3$, then G′ is an extremal graph, because if $J(G') < e(G') -2$ and β is any jump sequence of G′, then the concatenation $\nu = (\alpha(m_1), \beta)$ is a PEO of G with $J(\nu) \le e(G) - 3$. Let v''_1 be a simplicial vertex of G′ and let $L_{G'}(v_2) = m_2$ and α_1 be a PEO of G′ s.t. $\alpha_1[1] = v_2$ and $L_{G'}(\alpha_1) = m_2$. Let $\alpha_1(m_2)=(v''_1, v''_2, \dots, v''_{m_2})$. Define $G^2 = G^1 - \{ v''_1, v''_2, \dots, v''_{m_2} \}$.

Then $e(G^2) = e(G) - 2$. Also G^2 is extremal if $m > 4$. The above procedure gives a graph G^{m-1} s.t. $e(G^{m-1}) = 2$, which is H-Perfect by Lemma 5.4.6. Let α_{m-1} be an HEO of G^{m-1}. Then clearly $\alpha = (\alpha_1(m_1), \alpha_2(m_2), \ldots, \alpha_{m-2}(m_{m-2}), \alpha_{m-1})$ is a PEO of G and J(G)= m-2. Since G is extremal, α is a jump sequence of G starting from v_1. Hence the result.█

Proposition 5.4.8: The class of H-perfect graphs is not closed under vertex induced subgraphs.

Proof: Let $G_1^{(k)}$ be a p-k-tree with (k+3) vertices with simplicial vertices u, v, and w. Add two new vertices x and y to $G_1(k)$ and join x to all vertices but w, and y to all vertices but u. Then G(k) is H-perfect as $(w, y, v, x, u, s_1, s_2, \ldots, s_k)$ is an HEO of G(k), where $S = \{s_i, 1 \le i \le k\}$ is the set of vertices of $G_1^{(k)}$ of degree k+2. But $G_1^{(k)}$, an induced subgraph of G(k), is not H-perfect.█

In view of Proposition 5.4.8 it is not possible to characterize H-Perfect graphs in terms of minimal forbidden subgraphs. However, Theorem 5.4.10 gives a minimal forbidden subgraph characterization for H-perfect k-trees. Also there is no good relation between J(G) and e(G) for general chordal graphs. In Theorem 5.4.13 we prove that J(G) = e(G)-2, for every k-tree G.

Lemma 5.4.9: Let G be a k-tree. Let H be any induced non-complete sub k-tree of G. Then $|S(H)| \le |S(G)|$.

Proof: As H is non complete k-tree, $|S(H)| = e(H)$. Again G is constructible from H by Lemma 5.4.4. Let the order of H be n_1 and that of G be n. Let $\alpha = (v_1, v_2, \ldots, v_{n-n_1}, \ldots, v_n)$ be any constructing PEO of G w.r.t. H. Then by Lemma 5.4.6, $|S(G)| = e(G) \ge e(G-v_1) \ge e(G - \{v_1, v_2\}) \ge \ldots \ge e(G-\{v_1, v_2, \ldots, v_{n-n_1}\}) = e(H) = |S(H)|$. So $|S(H)| \le |S(G)|$.█

Theorem 5.4.10: Let G be a non-complete k-tree. Then the following are equivalent.

(a) G is H-Perfect.

(b) $|S(G)| = 2$.

(c) G has neither a p-k-tree with $(k+3)$ vertices nor a nearly p-k-tree with $(k+4)$ vertices as an induced subgraph.

(d) $e(G) = 2$.

Proof : (a) → (b)

Let G be an H-Perfect non-complete k-tree. We prove $|S(G)| = 2$ by induction on the order n of G. For $n = k+2$, $|S(G)| = 2$. Let us assume that $|S(G)| = 2$ for every H-Perfect k-tree G with fewer than n ($\geq k+3$) vertices and let G be an H-Perfect k-tree with n vertices. Let $(v_1, v_2, \ldots, v_n)$ be an HEO of G. Let $G'=G-v_1$. Now G' is a k-tree and it is noncomplete as $n \geq k+3$. Hence by induction principle $|S(G')| = 2$. since $v_1v_2 \in E$ and v_2 is a simplicial vertex of G', $|S(G)| = 2$.

(b) → (c)

Let G be k-tree with $|S(G)| = 2$. If possible, let G contain either a p-k-tree with $(k+3)$ vertices or a nearly p-k-tree with $(k+4)$ vertices as an induced subgraph. Let it be H. By Lemma 5.4.9, $3 = |S(H)| \leq |S(G)|$. So $|S(G)| \geq 3$, a contradiction.

(c) → (a)

Let G be a k-tree free from p-k-tree with $(k+3)$ vertices as well as a nearly p-k-tree with $(k+4)$ vertices. We claim that G is H-Perfect. To prove our claim we need the following intermediate result which we state as 'Fact'.

Fact: For any simplicial vertex v of G, $N(v)$ contains a simplicial vertex of $G'=G-v$.

Proof of the Fact:

Let v_1 be any simplicial vertex of G and let $N(v_1) = \{v_2, v_3, \ldots, v_{k+1}\}$. By Lemma 5.4.4, G can be constructed from $H = G[\{v_1,$

Proof : Let G be a k-tree and v be a simplicial vertex of G s.t. N(v) contains a simplicial vertex, say w, of $G' = G-v$. Let $S = N(v)$. Now as v is a simplicial vertex of G, $J(G) \geq J(G')$.......(1).

We claim that $J(G) \leq J(G')$. Let $\alpha = (v_1, v_2, \ldots, v_{n-1})$ be a jump sequence of G'. If $v_1 \in S$, then $\beta = (v, v_1, \ldots, v_{n-1})$ is a PEO of G with $J(\beta) = J(\alpha)$. So $J(G) \leq J(\beta) = J(\alpha) = J(G')$. So assume that $v_1 \notin S$. Let r be the largest index s.t. none of $v_1, \ldots, v_r$ belongs to S. So $v_{r+1} \in S$.

Case 1: $v_r v_{r+1} \notin E(G')$.

Let $\beta = (v_1, v_2, \ldots, v_r, v, v_{r+1}, \ldots, v_n)$. Then β is a PEO of G with $J(\beta) = J(\alpha) = J(G')$. So $J(G) \leq J(\beta) = J(G')$.

Case 2: $v_r v_{r+1} \in E(G')$.

Let $G'' = G[\{v_r, v_{r+1}, \ldots, v_{n-1}, v\}]$. We shall prove that the k-tree G'' has (k+2) vertices. If not, let $|V(G'')| \geq k+3$. As v and v_r are simplicial vertices of G'', v_{r+1} is a simplicial vertex of $G''-\{v, v_r\}$ and $v_r v_{r+1} \in E(G')$, $\deg_{G''}(v_{r+1}) = k+1$. As $v_{r+1} \in S$ and $v\, v_r \in E(G)$, $H = G''[(S \cup \{v, v_r\})]$ is a k-tree with k+2 vertices. As v and v_r are simplicial vertices of G'' and $\deg_{G''}(v_{r+1}) = k+1$, G'' cannot be constructed from H contradicting Lemma 5.4.4. Hence $|V(G'')| = k+2$.

Let $\beta = (v_1, v_2, \ldots, v_r, \ldots, v_{n-1}, v)$. Then clearly β is a PEO of G and $J(\beta) = J(\alpha)$. So $J(G) \leq J(\beta) = J(\alpha) = J(G')$, as α is a jump sequence of G'.

Therefore in either case $J(G) \leq J(G')$. Since $J(G) \geq J(G')$ by (1), $J(G) = J(G')$.∎

Theorem 5.4.13: Let G be a non-complete k-tree. Then $J(G) = e(G)-2$.

Proof : Our proof is by induction on $|S(G)|$. If $|S(G)| = 2$, then by Theorem 5.4.10, G is H-Perfect. So $J(G) = 0 = |S(G)| - 2$. Let us assume that the theorem is true for all non-complete k-trees with fewer than r ($r > 2$) simplicial vertices. Let G be a non-complete k-tree with r simplicial vertices. In view of Lemma 5.4.12, we may assume that N(v) does not contain

any simplicial vertex of G-v, for any $v \in S(G)$. Let v be a simplicial vertex of G and let $G' = G - v$. Now $|S(G')| = r-1$. Hence by induction principle $J(G') = e(G')-2 = r-3$. For every PEO $\alpha = (v, v_2, \ldots, v_n)$ of G $vv_2 \notin E(G)$ and $\beta=(v_2, v_3, \ldots, v_n)$ is a PEO of G'. As β is an arbitrary PEO of G', we have $J(G) = J(G') + 1 = r-2$. Hence by induction principle $J(G) = e(G) - 2$, for any non-complete k-tree.▌

We next design a linear time algorithm to find a jump sequence of a k-tree.

Algorithm A:

INPUT: A k-tree $G= (V,E)$ where $V = \{v_1, v_2, \ldots, v_n\}$.

OUTPUT: An ordering $\alpha= \{ w_1, w_2, \ldots, w_n\}$ s.t. $J(\alpha)=J(G)$.

METHOD:

```
BEGIN
  STEP 1: Define an array D by D[i] = deg(v_i);
          S:= { v_i∈ V | D[i]=k }; i:=1;
  STEP 2: Choose a vertex v_j ∈ S;v:= v_j;
  STEP 3: v_i:=v; S:=S-{v_i}; i:=i+1;
          If ( i < n-k-2) then
          begin
           for all v_j ∈ N(v_{i-1}) do
             begin
              D[j]:= D[j]-1;
              If D[j] = k then S:= S ∪ {v_j}
             end;
           If N(v_{i-1}) contains a vertex v_j with D[j] = k then
             begin
              v:= v_j; Go To STEP 3
             end
```

else go to STEP 2;

end;

STEP 4: For $i := n-k-2$ to n do

begin

choose a vertex $v \in S$;

$w_i := v$; $S := S-\{w_i\}$;

end;

END.

Theorem 5.4.14: The sequence α produced by Algorithm A is a PEO of G, and $J(\alpha) = J(G)$. Moreover, Algorithm A runs in $O(n+m)$ time.

Proof: Clearly the sequence α produced by Algorithm A is a PEO of G. Using the same argument as in the proof of Lemma 5.4.12, it can be proved that $J(\alpha) \le |S(G)| - 2$. Again by Theorem 5.4.13, $J(\alpha) \ge |S(G)| - 2$. So $J(\alpha) = |S(G)| - 2$.

It is easy to see that the Algorithm A runs in $O(n+m)$ time.▮

The following lemma whose proof follows from induction, is a key in obtaining a polynomial recognition algorithm for H-Perfect graphs.

Lemma 5.4.15: Let $\alpha = (v_1, v_2, \ldots, v_n)$ be an HEO of G. Let $N_{G_i}[v_i] = N_{G_i}[v_j]$, $i < j$, where $G_i = G[\{v_i, v_{i+1}, \ldots, v_n\}]$. Then $\alpha' = (v_1, v_2, \ldots v_{i-1}, v_j, v_{i+1}, \ldots, v_{j-1}, v_i, v_{j+1}, \ldots, v_n)$ is an HEO of G starting from v_1. Moreover, every HEO of G starting from v_1 can be obtained from α in a similar manner.

We now present a procedure HEO-FIND(v) which finds an HEO of G starting from a prescribed vertex v if such an HEO of G exists.

Procedure HEO-Find(v):

INPUT: A chordal graph G and a vertex v of G.

OUTPUT: An HEO $H[1], H[2], \ldots, H[n]$ of G where $H[1]=v$, if G has such an HEO.

METHOD:

```
BEGIN
    Step 1: i:=1; j:=2; v_1:=v; H[1]:=v_1;
    Step 2: While ( j not equal to n) do
        BEGIN
        (i) Find S(G);
        Let T:= { w_1, w_2,...,w_{l_r} } be the vertices in S(G) adjacent
        to v_i.
        If T = φ then Go To 10
        else
        begin
         t:=l_r; k:=1;
        end;
        (ii) For i:= j to j+t-1 do
            begin
             H[i]:= w_k; k:=k+1;
            end;
        (iii) j:=j+1; G:= G-T;
        End;
    10:     Declare "G does not have an HEO starting from v";
END.
```

The correctness of the Procedure HEO-FIND(v) follows from Lemma 5.4.15. It is easy to see that the above procedure runs in $O(n(n+m))$ time.

If G is H-Perfect then there exists a vertex v which is the starting vertex of some HEO of G. So applying the Procedure HEO-Find(v) $|S(G)|$ times, we can conclude whether G is H-Perfect. Moreover, if G is H-Perfect, we can obtain an HEO of G.

In view of the above discussion we have the following Theorem.

Theorem 5.4.16: H-Perfect graph can be recognized in $O(n^2m)$ time and for an H-Perfect graph an HEO can be constructed in same time bound.

The following theorem characterizes extremal chordal graphs.

Theorem 5.4.17: A chordal graph G with $e(G) = m+2$, $m > 2$ is extremal iff $G' = G - \alpha\,(L_G(v))$ is an extremal chordal graph with jump number $m-1$ for all $v \in S(G)$, where $\alpha[1]=v$.

Proof: Necessity:

If for some $v \in S(G)$, $G' = G - \alpha\,(L_G(v))$, $\alpha[1]=v$ is not extremal, then $J(G') < e(G') - 2 = m - 1$. Let β be any jump sequence of G'. Then $\alpha_1 = (\alpha\,(L_G(v), \beta)$ is a PEO of G with $J(\alpha_1) < m$. This contradicts the fact that G is extremal. Again once G' is extremal, $J(G') = e(G') - 2 = e(G) - 3 = m-1$.

Sufficiency:

If G is extremal, then there exists a PEO $\alpha = (v_1, v_2, \ldots, v_n)$ with $J(\alpha) < m$. Then $J(G') = J(G) - \alpha\,(L_G(v_1)) < m - 1$, which is a contradiction.∎

Theorem 5.4.18: A chordal graph G is extremal iff exactly one of the following conditions holds.

(a) For each $v \in S(G)$, $e(G-[v]) = e(G)-1$.

(b) There exists $v \in S(G)$ s.t. $e(G-[v]) = e(G)$ and $J(G-[v]) = e(G-[v])-2$.

Proof : As sufficiency is straightforward, we prove necessity only.

If (a) is not true, then there exists $v \in S(G)$ s.t. $e(G-[v]) = e(G)$. Let $[v] = \{v, v_1, v_2, \ldots, v_r\}$. As G is extremal there exists a jump sequence starting from v say $(v, v_1, v_2, \ldots, v_r, \ldots, v_{n-1})$. So $v_r v_{r+1} \in E$. Now $J(G-[v]) = J(G)$. So $J(G-[v]) = J(G)$. Thus (b) is true. It is easy to see that (a) and (b) are not simultaneously true. Hence the Theorem is true.∎

Note that Theorem 5.4.18 suggests a polynomial time recognition algorithm for extremal chordal graphs. For an extremal chordal graph G, one can design an $O(n^2m)$ algorithm to find a jump sequence of G, in the same line of Algorithm A.

5.5 Hamiltonian Chordal Graphs:

In spite of lots of efforts there is no non-trivial characterization of Hamiltonian graphs. In this section we characterize Hamiltonian H-Perfect graphs.

A pancyclic graph, i.e. a graph having cycle of all possible length, is necessarily Hamiltonian. For chordal graphs, the converse holds.

Theorem 5.5.1: A chordal graph G is Hamiltonian iff it is pancyclic.

Proof : Let G be a hamiltonian chordal graph of order n with $v_1, v_2, \ldots, v_n, v_1$ as a Hamiltonian cycle of G. Let v_i be a simplicial vertex of G. As $G[N(v_i)]$ is complete, $v_{i-1}v_{i+1} \in E$, where the indices are under modulo n. Now $G' = G-v_i$ is Hamiltonian as $v_1, v_2, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n, v_1$ is a Hamiltonian cycle of G′ of length n-1. This procedure yields cycles of every length starting from n through 3.∎

Corollary 5.5.2: Let G be a Chordal graph with circumference k. Then G has a cycle of length r, for $3 \le r \le k$.

THEOREM 5.5.3 : A chordal graph G is Hamiltonian iff it contains a Maximal outer planar (MOP) graph as a spanning subgraph.

Proof : Sufficiency : This follows from the fact that every MOP graph is Hamiltonian.

Necessity :

We claim that any Hamiltonian chordal graph G has a spanning MOP graph that contains all the edges of a given Hamiltonian cycle as its boundary edges. We prove our claim by induction on n, the order of G. For n = 3 and 4 our claim is trivially true. Assume that our claim is true for all Hamiltonian chordal graphs with fewer than n vertices. Let G be a Hamiltonian chordal graph of order n. Let $C = v_1, v_2, \ldots, v_n, v_1$ be any Hamiltonian cycle. Let v_i be any simplicial vertex of G. Then $G' = G-v_i$ is a Hamiltonian chordal graph and $C' = v_1, v_2, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n, v_1$ is a

Hamiltonian cycle of G'. So by induction hypothesis G' has a spanning MOP graph M' with C' as its boundary cycle. Then $M = M' \cup \{ v_i v_{i-1}, v_i v_{i+1} \}$ is a spanning MOP containing C as the boundary cycle. So by induction principle our claim is true. ∎

Since every MOP graph is a 2-Tree, with a unique Hamiltonian cycle, we have:

Corollary 5.5.4 [123]: A 2-Tree is Hamiltonian iff it is a MOP graph.

Corollary 5.5.5: The number of Hamiltonian cycles of a 2-Tree is at most 1.

Since the Hamiltonian Cycle Problem is NP-Complete for chordal graphs, by Theorem 5.5.3, We have the following result.

Theorem 5.5.6: The Problem of deciding whether a Chordal graph G has a MOP spanning subgraph is NP-complete.

We next focus our attention to the Hamiltonian cycle problem on H-Perfect Chordal graphs. We prove now two lemmas which we need for the characterization of H-Perfect Hamiltonian chordal graphs.

Lemma 5.5.7: Let $\alpha = (v_1, v_2, \ldots, v_n)$ be an HEO of an H-Perfect chordal graph. If $v_i v_j \in E$, then $v_k v_j \in E$ for all k, $i \le k \le j-1$.

Proof: As α is an HEO, $v_i v_{i+1} \in E$. Given that $v_i v_j \in E$. As v_i is a simplicial vertex of $G_i = G[\{v_i, v_{i+1}, \ldots, v_n\}]$, $G[N_{G_i}(v_i)]$ is complete. So $v_{i+1} v_j \in E$. Again v_{i+1} is a simplicial vertex of $G_{i+1}[\{ v_{i+1}, \ldots, v_n\}]$. So $v_{i+2} v_j \in E$. By the same argument, $v_k v_j \in E$, $i \le k \le j-1$. ∎

Lemma 5.5.8: Let G be a biconnected, H-Perfect chordal graph and let $\alpha = (v_1, v_2, \ldots, v_n)$ be an HEO of G. Then $d_{G_i}(v_i) \ge 2$, $1 \le i \le n-2$, where $G_i = G[\{v_i, v_{i+1}, \ldots, v_n\}]$.

Proof: If possible, let i be the largest index s.t. $d_{G_i}(v_i) < 2$. Then, as $v_i v_{i+1} \in E$, $d_{G_i}(v_i) = 1$. Again $1 < i < n-1$. We claim that v_{i+1} is a cut vertex of G. If $v_j v_k \in E$, for $j \le i$ and $k \ge i+2$, then by Lemma 5.5.7, v_{j+s}

$v_k \in E$, for all s, $1 \le s \le k-j-1$. In particular, $v_i v_k \in E$. This contradicts $d_{G_i}(v_i)=1$. So $v_j v_k \notin E$ for $j \le i$ and $k \ge i+2$. So every path from v_j to v_k, $j \le i$ and $k \ge i+2$, passes through v_{i+1}. Thus our claim is true. This contradicts the biconnectedness of G, and completes the proof.∎

Theorem 5.5.9: Let G be an H-Perfect graph. Then G is hamiltonian iff G is biconnected.

Proof: Since the necessity is trivial we prove the sufficiency only.

Let $\alpha = (v_1, v_2, \ldots, v_n)$ be an HEO of G. Let i_1 be the largest index s.t. $v_1 v_{i_1} \in E$. Let i_2 be the largest index s.t. $v_{i_1 -1} v_{i_2} \in E$. Let i_j be the largest index s.t. $v_{i_{j-1}-1} v_{i_j} \in E$. Let r be the smallest index s.t. $v_{i_r -1} v_n \in E$, but $v_{i_j -1} v_n \notin E$ for $1 \le j \le r-1$. As G is biconnected H-Perfect chordal graph, $i_j - 1 > i_{j-1}$ and $i_{r-1} \le n-2$. Let P_j be the path $v_{i_{j-1}-1} v_{i_j}$ consisting of only one edge, $2 \le j \le r+1$, where $v_{i_{r+1}} = v_n$. Let Q_j be the path from v_{i_j} to $v_{i_{j+1}-1}$ through consecutive vertices of the HEO α, $1 \le j \le r-1$. Let P_1 be the path $v_1 v_{i_1}$ consisting of only one edge. Let Q_1' be the path from v_1 to $v_{i_1 -1}$ through consecutive vertices. Let Q_r be the path from v_{i_r} to v_n through consecutive vertices of the HEO α. Then $C = Q_1' \cup (\bigcup_{j=1}^{r+1} P_j) \cup (\bigcup_{j=1}^{r} Q_j)$ is a Hamiltonian cycle of G, whence G is Hamiltonian.∎

Given an H-Perfect chordal graph and an HEO α, we next propose a linear time algorithm to recognize whether the given graph is Hamiltonian, and if so we construct a Hamiltonian cycle.

Algorithm Hamiltonian Cycle:

Input: $\{G, \alpha = (v_1, v_2, \ldots, v_n)$, where G is an H-Perfect chordal graph and α is an HEO of G.

Output: If G is non Hamiltonian, then output ' G is not Hamiltonian' else output 'G is Hamiltonian' and a Hamiltonian cycle C.

Method:

```
Begin
    If G is not biconnected then output 'G is not Hamiltonian'
Else
Begin
        C := v_1, v_2,..., v_n ; t := 1; r :=1;
        While ( t ≠ n) do
            Begin
              Let l_r be the largest index s.t. v_r v_{l_r} ∈ E;
              If ( l_r ≠ n) Then
                  Begin
                    C := C - { v_{l_r-1} v_{l_r} } ∪ { v_r v_{l_r} };
                    r := l_r-1;
                  End
              Else
                  C := C ∪ { v_r v_{l_r} }; t:= l_r;
            End;
    End
End.
```

It is easy to see that algorithm Hamiltonian Cycle runs in O(n+m) time. The correctness of the above algorithm follows from Theorem 5.5.9. So we have the following result.

Theorem 5.5.10: 'Algorithm Hamiltonian Cycle' correctly finds in O(n+m) time a Hamiltonian cycle of an H-Perfect graph G if G is Hamiltonian otherwise reports that G is non Hamiltonian.

5.6 Hamiltonian Cycles In Proper Interval Graphs:

The problem of deciding whether a graph has a Hamiltonian cycle is well known to be NP-complete[47], and remains so for several special classes of graphs, such as planar cubic 3-connected graphs[48], bipartite graphs[74], split graphs[57], edge graphs[12], planar bipartite graphs[48], grid graphs [74], undirected path graphs[14], double interval graphs[14], and rectangle graphs[14]. In contrast, polynomial algorithms for solving this problem exist only for very restricted classes of graphs, such as planar 4-connected graphs[61], proper circular graphs[13], interval graphs[77], bipartite distance-hereditary graphs[94], and circular arc graphs[120].

A graph G is said to be t-tough if after the deletion of an arbitrary set of s vertices, the graph remains connected or has at most $\frac{s}{t}$ components. Chvatal's [24] conjecture that every 2-tough graph is Hamiltonian is yet to be settled.

A. Bertossi[13] suggested an O(n log(n)) time algorithm for the Hamiltonian cycle problem in a proper interval graph given in interval representation. But to find an interval representation of a proper interval graph which is given in adjacency list representation, takes O(n+m) time. So his algorithm takes O(n + m + nlog(n)) time if the graph is given in adjacency list representation. Later J. M. Keil [77] suggested an O(n+m) time algorithm for the Hamiltonian cycle problem for interval graphs. In this section we suggest a linear time algorithm for finding Hamiltonian cycles in proper interval graphs. Our algorithm is as good as the algorithm that follows from Keil's [77] work as far as the complexity of the algorithm is concerned. Unlike keil's [77] algorithm, our algorithm is supported by good theoretical characterization which has other implications too. In fact, we have proved in the last section that a proper interval

graph is Hamiltonian iff it is biconnected, since every connected proper interval graph is H-perfect. since every 2-tough graph is biconnected, chvatal's above mentioned conjecture holds good for H-perfect chordal graphs and in particular for proper interval graphs.

To obtain a linear time algorithm for Hamiltonian cycle problem for proper interval graphs, we proceed as follows:

we introduce the notion of strong simplicial vertex, and show that every proper interval graph has exactly two strong simplicial vertices up to certain equivalence(Theorem 5.6.1). An algorithm for finding the two such vertices of a proper interval graph is given (see Theorem 5.6.4). We then present an algorithm for finding an HEO of a chordal graph having exactly two simplicial vertices. This is done because it can be used to find an HEO α of a proper interval graph G by first finding an HEO α' of $G'=G-(S(G)-\{u,v\})$, where S(G) is the set of simplicial vertices of G, and u, and v are the strong simplicial vertices of G, and then incorporating the vertices of $S(G)-\{u,v\}$ to α' suitably. Finally the algorithm in section 5 can be used to find a Hamiltonian cycle in G, as every proper interval graph is H-perfect.

A simplicial vertex v of a graph G is said to be a strong simplicial vertex if there exists a simplicial vertex w of G' adjacent to v, where $G'=G-[v]$ and [v] is the simplicial class of S(G) containing v.

Note that if $(v_1, v_2,\dots,v_n)$ is an HEO of G, then v_1 is a strong simplicial vertex of G.

Two strong simplicial vertices v and w are said to be equivalent if $[v] = [w]$.

Theorem 5.6.1: Every proper interval graph G has exactly two strong simplicial vertices up to equivalence.

In order to prove Theorem 5.6.1 we need the following Lemma.

Lemma 5.6.2: Let $G_i=G[V_i \cup C]$, $1\le i\le r$, $r\ge 2$, be the separated subgraphs of a proper interval graph w.r.t. any separating clique C of G. Then r = 2.

Proof: If not, let $r \ge 3$. Let $W(G_i)= \{s \in C$ s.t. there exist v in V_i-C with $vs \in E\}$. Since G is Chordal, there exist C_i in G_i s.t. $W(G_i) \subset C_i$. Then $G'=G[\{C_1\cup C_2\cup C_3\cup C\}]$ is a Proper interval graph. Also $(C_i\cap C) \ne \phi$.

Case 1: There exist i,j with $W(G_i) \subseteq W(G_j)$, $1 \le i,j \le 3$.

Now $G'[\{v_i,v_j, v',v''\}]$ is isomorphic to $K_{1,3}$, where $v_i\in C_i-C$, $v_j\in C_j-C$, $v'\in C_i\cap C$, and $v'' \in C-C_j$, a contradiction to Theorem 1.4.7.

Case 2: There exist no i, j with $W(G_i) \subset W(G_j)$.

Let C_{i_1}, C_{i_2}, C_{i_3}, C_{i_4} be any linear ordering of C_1, C_2, C_3,C. So either at least two cliques lie to the left of C or at least two cliques lie to the right of C. Wlg, Let C_{i_1} and C_{i_2} lie to the left of C in the above linear ordering. Let $x \in (C_{i_1} \cap C) -(C_{i_2} -C)$. The existence of x is assured by Case 2. Now x lies in C_{i_1} and C but not in C_{i_2}. So the Cliques of G′ cannot be ordered in such a way that the set of maximal cliques containing a vertex occur consecutively, a contradiction to Theorem 1.4.5.∎

Proof of Theorem 5.6.1: We induct on k, the number of maximal cliques of G. If k=2, our Lemma is easily seen to be true. Let G be a proper interval graph with k+1 cliques. Since k +1 >2, G has a separating clique, say C. By Lemma 5.6.2, there will be exactly two separated graphs of G w.r.t. C, say G_1 and G_2. Since G is a proper interval graph, by Theorem 1.4.9, it has a BCO, say, $(v_1, v_2,\ldots, v_n)$. Wlg, let $v_1 \in V(G_1)$. Let $|C| = k$ and $|V(G_1)| = r$. Then $(v_1, v_2,\ldots,v_r)$ is a BCO of G_1 s.t. v_r, $v_{r-1},\ldots,v_{r-k+1}$ are the vertices from C. Again, by induction hypothesis, G_1 and G_2 satisfy our Lemma. With the notation as above, let v_1 and v_r be the only two strong simplicial vertices of G_1. Since $(v_n, v_{n-1},\ldots,v_1)$ is again a BCO of G, $(v_n, v_{n-1},\ldots, v_{r-k+1})$ is a BCO of G_2. So v_n and v_{r-k+1} are the only strong

simplicial vertices of G_2. Since no strong simplicial vertex belongs to C, v_1 and v_n are the only strong simplicial vertices of G.▌

We now present a Procedure to find the two strong simplicial vertices of a proper interval graph.

Procedure A1:

Input: A proper interval graph G s.t. S(G) is independent.

Output: Two strong simplicial vertices w_1 and w_2 of G.

Method:

```
Begin
  STEP 1: Find S(G); Let S(G)={ v_1, v_2,...,v_k};
          For i:=1 to  n do
          A[v_i]:=j if v_i lies in exactly j maximal cliques        of G;
  STEP 2: G':=G-S(G); Find S(G'); Let S(G')={v'_1,v'_2,...,v'_r};   s:=1;
  STEP 3: For i:=1 to r do
          Begin
           T(v'_i):={ v_i s.t. v_i∈ S(G) and A[v_i]=1};
           C(v'_i):=|T(v'_i)|; (* |T(v'_i)| is the size of T(v'_i) *)
           If C(v'_i)= 1 Then
            Begin
            w_s := v_i, where {v_i}= C(v'_i); s:=s+1 else
             If C(v'_i)= 2 Then
             Begin
               Let v_1, v_2 ∈ C(v'_i); If d(v'_i)-1 = d(v_1) Then
               (i) w_s:= v_2 (ii) s:=s+1; Else
               If d(v'_i)-1=d(v_2) Then
               (i) w_s:= v_1; (ii) s:= s+1;
              End;
            End;
```

End.

In order to prove the correctness of the Procedure A1, we need the following result.

Lemma 5.6.3: Let G be a proper interval graph s.t. S(G) is independent. Let G'=G-S(G). Then every $v \in S(G')$ is adjacent to at most two vertices of S(G). If v is adjacent to exactly one vertex, say v_1 of S(G), then v_1 is a strong simplicial vertex of G. If v is adjacent to two vertices, say v_1 and v_2 of S(G), then v_1 is a strong simplicial vertex iff $d(v)-1 = d(v_2)$, v_2 is a strong simplicial vertex iff $d(v)-1=d(v_1)$, and exactly one of v_1 and v_2 must be a strong simplicial vertex in this case, unless G has exactly two maximal cliques.

Proof: If any vertex of $S(G')$ is adjacent to three or more vertices of S(G), then G will contain a $K_{1,3}$ as S(G) is independent, which is a contradiction to Theorem 1.4.7. If v is adjacent to exactly one vertex of S(G), say v_1, then clearly v_1 is a strong simplicial vertex of G as S(G) is independent. Let a vertex v of $S(G')$ be adjacent to two vertices v_1 and v_2 of S(G). Since v_1 and v_2 are simplicial vertices of G and v is in $N(v_i)$, i = 1,2, $N(v_i) \subseteq N(v)$. So $d(v_i) \leq d(v)$, i = 1,2. If $d(v) -1=d(v_2)$, then clearly v_1 is a strong simplicial vertex of G. Similarly if $d(v)-1 = d(v_1)$, then v_2 is a strong simplicial vertex of G. If neither $d(v)-1=d(v_1)$ nor $d(v)-1= d(v_2)$, then G will contain a $K_{1,3}$, which is absurd by Theorem 1.4.7. Again if possible, let v_1 and v_2 be both strong simplicial vertices. If G has more than two maximal cliques, then there exist w_1 and w_2 s.t. $w_1 \in N(v)-\{v_1, v_2\}$, $w_2 \notin N(v)$ and $w_1w_2 \in E$. Then $G[\{w_1,w_2,v_1,v_2\}]$ is isomorphic to $K_{1,3}$, again a contradiction to Theorem 1.4.7.∎

Theorem 5.6.4: Procedure A1 Correctly finds two strong simplicial vertices of a proper interval graph G having S(G) independent.

Proof: The existence of two strong simplicial vertices is assured by

Theorem 5.6.1 and the correctness of Procedure A1 follows from Lemma 5.6.3.∎

We now suggest a procedure to find an HEO of a chordal graph with exactly two simplicial classes.

Procedure A2:

Input: A connected chordal graph G with exactly two simplicial classes.

Output: An array H[1..n] s.t. H[1], H[2],...,H[n] is an HEO of G.

Method:

```
Begin
STEP 1: Find the maximal cliques of G. Let C_{i_1}, C_{i_2},...,C_{i_r} be the
        maximal cliques of G.
STEP 2: Compute an array A s.t. A[v_i]=j if v_i lies in exactly j
        cliques of G.
STEP 3: S(G):= {v_i s.t. A[v_i]=1 }. For all v ∈ V, M(v)=0.
STEP 4: i:=2; x:=v_j for some v_j ∈ S(G); H[1]=x; M(H[1]):=-1;
        While ( i ≠n+1) do
        Begin
          For all v_k ∈ ADJ(x) do
              If ( A[v_k]=1 and M(v_k)=0) Then
                Begin
                  x:=v_k; H[i]:=x; M(H[i]):=-1; i:=i+1;
                End;
              For all v_s∈ ADJ(x) do
                  If A[v_s] ≥ 1 Then A[v_s] := A[v_s]-1;
            End;
End.
```

The following result will be used to prove the correctness of Procedure A2.

Lemma 5.6.5: Let G be a chordal graph and [v] be a simplicial class of S(G). If N(v)-[v] is a maximal clique of G'=G-[v], then N[v] is a separating clique of G.

Proof: Let C= N(v)-[v]. Let x and y be two vertices of G' s.t. $N_{G'}(x) \cap C$ and $N_{G'}(y) \cap C$ are maximal sets. Existence of such vertices is assured as C is a maximal clique of G' and C = N(v)-[v] Let G" = G-N[v]. If x and y are connected in G" then let P be a shortest x --y path in G". Since $N_{G'}(x) \cap C \neq N_{G'}(y) \cap C$, there exist $z_1 \in N_{G'}(x)-N_{G'}(y)$ and $z_2 \in N_{G'}(y)-N_{G'}(x)$. Then P $\cup$ $\{z_1x, z_2y\}$ will contain a chordless cycle of length at least 4, a contradiction. So there is no x--y path in G". Hence N[v] is a separating clique of G.▌

Theorem 5.6.6: Procedure A2 correctly finds an HEO of a chordal graph having exactly two simplicial classes.

Proof: Since H[1], H[2],...,H[n] is a Hamiltonian path, we need only to prove that in every iteration of the while loop in Step 4 of Procedure A2, $A[v_i]=j$ iff v_i lies in exactly j cliques of the subgraph of G induced by the set of vertices having M(.)=0, in that iteration. If possible, let i be the smallest index s.t. our claim is not true at the end of the ith iteration. Consider the value of x at the end of the ith iteration. Since our claim is not true at the end of the ith iteration, C'=M[x]-[x] is a maximal clique of G", where V(G")= { y s.t. M(y)=0 at the end of the ith iteration }, and [x] is a simplicial class of G', where $V(G')$= {y s.t. M(y)=0 at the starting of the ith iteration}. By Lemma 5.6.3, C' is a separating clique of G". Let G_1'' and G_2'' be two separated graphs of G". Let C be a maximal clique of G^* containing [x], where $V(G^*)$=[x] $\cup$ { y s.t.t M(y) = -1 at the starting of the ith iteration }. Then C is a separating clique of G. Hence G has at least three simplicial classes as $W(G_1") \cup W(G_2") \neq C$, a Contradiction !▌

We are now in a position to suggest an algorithm for finding an HEO of a general proper interval graph.

Algorithm HEO:

Input: A proper interval graph G.

Output: An HEO of G.

Method:

Begin

STEP 1: Find S(G). Find the simplicial classes of S(G).

Name the simplicial classes by the vertex having the least index. Let the simplicial classes be

$[v_{i_1}], [v_{i_2}], \ldots, [v_{i_r}]$.

$G' := G - (S(G) - \{v_{i_1}, v_{i_2}, \ldots, v_{i_r}\})$;

STEP 2: Find the two strong simplicial vertice of G′ using procedure A 1; Wlg, let v_{i_1}, and v_{i_2} be the strong simplicial vertices of G′.

$G'' := G' - \{v_{i_3}, v_{i_4}, \ldots, v_{i_r}\}$;

STEP 3: Find an HEO α″ of G″ using procedure A 2 starting from v_{i_1};

Let $\alpha'' = (w_1, w_2, \ldots, w_{n_1})$; let $k_1, k_2, \ldots, k_r$ be s.t.

$\{w_{k_1}, w_{k_2}, \ldots w_{k_r}\} = \{v_{i_1}, v_{i_2}, \ldots, v_{i_r}\}$ and $k_1 < k_2 < \ldots < k_r$.

Let $\alpha_{k_j} = (\alpha_{k_j}(1), \alpha_{k_j}(2), \ldots, \alpha_{k_j}(s_k))$ be an ordering of $[v_{k_j}]$ s.t. $|[v_{k_j}]|$, the size of $[v_{k_j}]$, is s_k;

$\alpha = (\alpha_{k_1}(1), \alpha_{k_1}(2), \ldots, \alpha_{k_1}(s_1), w_2, \ldots, w_{k_j-1}, \alpha_{k_j}(1),$
$\alpha_{k_j}(2), \ldots, \alpha_{k_j}(s_j), w_{k_j+1}, \ldots, w_{k_r-1}, \ldots, \alpha_{k_r}(1), \ldots,$
$\alpha_{k_r}(s_r), w_{k_r+1}, \ldots, w_{n_1})$;

(* Incorporate all the vertices of $[w_{k_j}]$ but w_{k_j} just

after w_{k_j}, $1 \le j \le k$, in the sequence α'' and obtain α);

End.

It is easy to see that Algorithm HEO correctly finds an HEO of a proper interval graph. Since the set of maximal cliques of a chordal graph can be computed in O(n+m) time [49,57], and since $(\sum(|C|,\ C \in \mathcal{C}(G)) = O(n+m)$, for a chordal graph[57], it is not difficult to see that Algorithm HEO takes O(n+m) time. Since proper interval graphs are H-perfect, we can find a Hamiltonian cycle in a proper interval graph in O(n+m) time by using the Algorithm HEO, and Algorithm Hamiltonian Cycle of the last section.

In view of the above discussion we have:

Theorem 5.6.7: Hamiltonian proper interval graphs can be recognized in O(n+m) time. In a Hamiltonian proper interval graph, a Hamiltonian cycle can be constructed in O(n+m) time.

REFERENCES

1. A.V.Aho, J.E.Hopcroft, and J.D.Ullman, The Design and Analysis of Computer Algorithms, Addison Wesley, Reading, Massachusetts, 1974.

2. A.V.Aho, J.E.Hopcroft, and J.D.Ullman, Data Structure and Algorithm, Addison Wesley, Reading, 1983.

3. S.G.Akl, The Design and analysis of Parallel Algorithm, Prentice Hall, Englewood cliff, New Jersey, 1990.

4. L.Beineke and R. Pippert, Properties and characterizations of k-trees, Mathematica, 18 (1971) 141-151.

5. C.Berge, Les Problemes de colorations en theorie des graphs, Publ. Inst. Statist. Univ. Paris, 9 (1960) 123-160.

6. C.Berge, Farbung von Graphen, deren samtliche bzw. deren ungerade Kreise starr sind, wiss Z. Martin-Luther-Univ. Halle-Wittenberg Math-Natur, (1961) Reihe, 114-115.

7. C.Berge, Sur Une conjecture relative au probleme des codes optimaux, Comm. 13ieme Assemblee Gen. URSI,Tokyo,1962.

8. C.Berge, Some Classes of perfect graphs, in: Graph Theory and Theoretical Physics, Academic Press, (1967) 155-165.

9. C.Berge, Perfect Graphs, in: Studies in Graph Theory, D.R.Fulkerson Ed. Part-I, 1-22, 1975.

10. C.Berge, Theory of Graphs, (North-Holland, Amsterdam, 1984).

11. C.Berge and V.Chvatal (Eds.), Topics on Perfect graphs, Ann. of Discrete Math. 21,(1984).

12. A.A.Bertossi, The edge Hamiltonian path problem is NP-complete I.P.L. 13 (1981) 157-159.

13 A.A.Bertossi, Finding Hamiltonian circuits in proper interval graphs, I.P.L., 17 (1983), 97-101.

14. A.A. Bertossi and M.A. Bonuccelli, Hamiltonian circuits in interval graph generalizations, I.P.L. 23 (1986) 195-200.

15. R.G.Bland, H.C.Huang, and L.E.Trotter Jr., Graphical properties related to minimal imperfection, Discrete Math. 27(1979) 11-22.

16. J.A.Bondy and U.S.R.Murthy, Graph Theory With Applications, The Macmillan Press London, 1976.

17. K.S.Booth and G.S.Leuker, Testing for the consecutive ones property, Interval graphs, and graph planarity using PQ-tree algorithm, J. COMP. SYST. SCI 13 (1976) 335-379.

18. P. Buneman, A characterization of rigid circuit graphs, Discrete Math. 9 (1974) 205-212.

19. P.Buneman, The recovery of trees from measures of dissimilarity, Mathematics in the Archaeological and Historical sciences, Edinburg Univ. Press, Edinburg (1972) 387-395.

20. O.M.Carducci, The Strong Perfect Graph Conjecture holds for diamonded odd cycle-free graphs, Discrete Math. 110 (1992) 17-34.

21. N.Chandrasekharan and S.S.Iyenger, NC Algorithms For Recognizing Chordal Graphs and k-trees, IEEE Tras. Comp. 37 (1988) 1178-1183.

22. R. Chandrasekharan and A. Tamir, Polynomially bounded algorithms for locating p-centers on a tree, Math. Programming, 22 (1982) 304-315.

23. Christofides, Graph Theory: An Algorithmic Approach, Academic Press, 1975.

24. V.Chvatal, Tough graphs and Hamiltonian circuits, Discrete Math 5 (1973) 215-228.

25. V.Chvatal, On the strong perfect graph conjecture, JCT B 20 (1976) 139-141.

26. V.Chvatal, R.L.Graham, A.F.Perold and S.H.Whitesides, Combinatorial designs related to the strong perfect graph conjecture, Discrete Math. 26 (1979) 83-92.

27. V.Chvatal, Star-cutsets and Perfect Graphs, JCT B 39 (1985) 189-199.

28. R.Cole, Parallel Merge Sort, SIAM J COMP 17 (1988) 770-785.

29. S.Cook, The classification of Problems which have fast parallel Algorithms, FCT 1983.

30. D.G.Corneil, Families of Graphs complete for the strong perfect graph conjecture, J. Graph Theory 10 (1986) 33-40.

31. D.G.Corneil, Y.Peri, and L.stewart Burlingham, A linear time recognition algorithm for Co graphs, SIAM. J. Comput. 14 (1985), 926-934.

32. P.F.Deitz, Intersection graph Algorithms, TR 84-628 (Ph.D. Thesis), Comp. Sci. Deptt., Cornell University, 1984.

33. N.Deo, Graph Theory with applications to engineering and computer sciences, Prentice-Hall of India Pvt. Ltd. 1987.

34. D. de werra, On line perfect graphs, Math. Program. 15 (1978) 236-238.

35. G.A.Dirac, On rigid circuit graphs, Abh. Math. sem univ. Hamburg 25 (1961) 71-76.

36. P.Duchet, Classical perfect graphs, in: Topics in perfect graphs, C.Berge and V. Chvatal Eds., Ann. Discrete Math. 21 (1984) 67-96.

37. A.Edenbrandt, Chordal Graph Recognition is in NC, IPL 24 (1987) 239-241.

38. S.Even, Graph Algorithm, Computer Science Press, Inc., 1979.

39. R.Fagin, Degrees of acyclicity for hypergraphs and relational database schemes, J.A.C.M. 30 (1983) 514-550.

40. M. Farber, Characterization of strongly chordal graphs, Discrete Math. 43 (1983) 173-189.

41. D.Foata, Enumerating k-trees, Discrete Math. 1 (1972) 181-186.

42. S.Fortune and J.Wyllie, Parallelism in random access machines, STOC (1978) 114-118.

43. D.R.Fulkerson, The Perfect graph conjecture and Pluperfect graph Theorem, 2nd chapel Hill Conf. on Comb. Math. and its Appl. (1969) 171-175.

44. D.R.Fulkerson, Blocking and antiblocking pairs of polyhedra, Math. Prog. 1 (1971) 168-194.

45. D.R.Fulkerson, Antiblocking polyhedra, JCT B 12 (1972) 50-71.

46. D.R.Fulkerson and O.A.Gross, Incidence matrices and Interval graphs, Pacific J. Math. 15 (1965) 835-855.

47. M.R.Garey and D.S.Johnson, Computers and Intractability: A guide to the theory of NP-completeness, (W.H.Freeman and Co., San francisco), 1979.

48. M.R. Garey, D.S. Johnson, and R.E. Tarjan, The planar Hamiltonian cycle problem is NP-complete, SIAM J. Comput. 5 (1976) 704-714.

49. F.Gavril, Algorithms for minimum coloring, maximum cliques, minimum covering by cliques and maximum independent set in a chordal graph, SIAM J. Comput. 1 (1972) 180-187.

50. F.Gavril, An Algorithm for testing chordality of graphs, I.P.L. 3 (1974) 110-112.

51. F. Gavril, The intersection graphs of subtrees in trees are exactly the chordal graphs, JCT B 16 (1974), 47-56.

52. F. Gavril, A recognition algorithm for the intersection graphs of directed paths in directed trees, Discrete Math. 13 (1975) 237-249.

53. F. Gavril, A recognition algorithm for the intersection graphs of paths in trees, Discrete Math. 23 (1978) 211-227.

54. A.M.Gibbons, Algorithmic Graph Theory, Cambridge University Press, 1985.

55. A.Gibbons and W.Rytter, Efficient Parallel Algorithms, Cambridge University Press, Cambridge, 1988.

56. P.C. Gilmore and A.J.Hoffman, A characterization of comparability graphs and of interval graphs, Canadian J. Maths. 16 (1964) 539-548.

57. M.C.Golumbic, Algorithmic graph theory and perfect graphs, Academic press, New York 1980.

58. M.C.Golumbic (Ed.), Interval Graphs and related Topics, Discrete Math. 55 (1985).

59. M.C.Golumbic, Algorithmic aspects of intersection graphs and representing Hypergraphs, Graphs and Combinatorics, 4 (1988) 307-321.

60. M. C. Golumbic and R. E. Jamison, The edge intersection graphs of paths in a tree, JCT B 38 (1985), 8-22.

61. D. Gouyou-Beauchamps, The Hamiltonian circuit problem is polynomial for 4-connected planar graphs, SIAM J. Comput. 11 (1982) 529-539.

62. C.M.Grinsted, The perfect graph conjecture for torodial graphs, JCT B 30 (1981) 70-74.

63. M.Grotschel, L.Lovasz and A.schrijver, Polynomial algorithms for perfect graphs, Ann. Dis. Math. 21 (1984) 325-356.

64. M.Grotschel, L.Lovasz, and A.Schrijver, Geometric Algorithms and Combinatorial Optimization, Springer-Verlag, 1988.

65. A.Hajnal and J.Suranyi, Uber die auflosung von graphen in vollstandige teilgraphen, Ann. Univ. Sci. Budapest Eotvos, Sect. Math. 1 (1958) 113-121.

66. F.Harary, A characterization of block graphs, Canad. Math. Bull. 6 (1963) 1-6.

67. F.Harary, Graph Theory, Adission-Wesley, Reading, MA, 1969.

68. F.Harary and E.Palmer, On acyclic simplicial complexes, Mathematica, 15 (1968) 115-122.

69. D.A.Holton, A report on stable graphs, J. Austral. Math. Soc. 15 (1973) 163-171.

70. D.A.Holton, Two applications of semistable graphs, Discrete Math. 4 (1973) 151-158.

71. D.A.Holton, Stable Trees, J. Austral. Math. Soc A 15 (1973) 476-481.

72. D.A.Holton and D.D.Grant, Regular Graphs and Stability, J. Austral. Math. Soc. A 20 (1975) 377-384.

73. E.Horowitz and S.Sahni, Fundamentals of Computer Algorithms, Computer Science Press, 1984.

74. A.Itai, C.H. Papadimitriou, and J.L. Szwarcfiter, Hamiltonian paths in grid graphs, SIAM J. Comput. 11 (1982) 676-686.

75. J.JA'JA and J.Simon, Parallel Algorithms in Graph Theory: Planarity Testing, SIAM J Comput. 40 (1982) 314-328.

76. R.E.Jamison and R.Laskar, Elimination orderings of chordal graphs, in: Proc. of the seminar on combinatorics and applications (K. S.Vijain et al (eds.)) I.S.I.Calcutta (1982) 192-200.

77. J.M.Keil, Finding Hamiltonian circuits in interval graphs, IPL 20 (1985) 201-206.

78. V. Klee, What are the intersection graphs of arcs in a circle ?, Amer. Math. Monthly 76 (1969) 810-813.

79. N.Korte and R.H.Mohoring, An incremental linear-time Algorithm for recognizing Interval graphs, SIAM J. Comput. 18 (1989) 68-81.

80. P.N.Klein, Efficient Parallel Algorithms For Chordal Graphs, in: Proc. 29th Annual IEEE Symposium on Foundations of Computer Sciences (1988), 150-161.

81. P.N.Klein and J.Reif, An efficient parallel algorithm for planarity, JCSS (1988) 190-246.

82. D.E.Knuth, The Art of Computer Programming Vol-1: Fundamental Algorithms, Add-Wesley, Reading, MA 1973.

83. Van Bang Le, Perfect k-line graphs and k-total graphs, J.Graph Theory 17 (1993) 65-73.

84. C.G.Lekkerkerker and J.C. Boland, Representation of a finite graph by a set of intervals on the real line, 51 (1962) 45-64.

85. W.A.Lobb, Perfect graphs from paths in trees, unpublished manuscript.

86. L.Lovasz, Normal Hypergraphs and the Perfect graph conjecture, Discrete Math. 12 (1972) 253-267.

87. L.Lovasz, A characterization of Perfect Graphs, JCT B 13 (1972) 95-98.

88. L.Lovasz, Perfect Graphs, in: Wilson and Beineke Eds., Selected topics in Graph Theory-2, Academic press(1983) 55-87.

89. A.Lubiw, Short-chorded and Perfect graphs, JCT B 51(1991) 24-33.

90. Mchugh, Algorithmic Graph Theory, Prentice Hall, 1990.

91. H.Meyniel, On the perfect graph conjecture, Discrete Math. 16 (1976) 339-342.

92. C.L.Monma and V.K.Wei, Intersection graphs of paths in a tree, JCT B 41 (1986) 141-181.

93. A.Moitra and S.S. Iyenger, Parallel Algorithms for some computation problems, Adv. Comput. 26(1987) 95-153.

94. H.Muller and F.Nicolai, Polynomial time Algorithms for Hamiltonian Problems on bipartite distance-hereditary graphs, IPL 46 (1993) 225-230.

95. J.Noar, M.Noar, and A.Schaffer, Fast parallel algorithms for Chordal graphs, SIAM J Comput. 18 (1989)327-349.

96. S.Olariu, The Strong perfect graph conjecture for pan-free graphs, JCT B 47 (1989) 187-191.

97. C.Papadimitriou and M.Yanakakis, Scheduling Interval ordered Tasks, SIAM J. Comput. 8 (1979) 405-409.

98. K.R.Parthasarathy and G.Ravindra, The SPGC is true for $(K_{1,3})$-free graph, JCT B 21(1976) 212-223.

99. K.R.Parthasarathy and G.Ravindra, The validity of SPGC for $(K_4$-e) free graphs, JCT B 26 (1979) 98-100.

100. M.Preismann, Locally Perfect Graphs, JCT B 50 (1990) 22-40.

101. A.Proskurowski, k-trres: representations and distances, Tech. Rep., CIS-TR-80-5, Univ. Oregon, OR, 1980.

102. A.Proskurowski, Recursive graphs, recursive labellings and shortest paths, SIAM J. Comput. 10 (1981) 391-397.

103. A.Proskurowski, Separating subgraphs in k-trees, cables and caterpillars, Discrete Math. 49 (1984) 275-285.

104. M.J.Quinn and N.Deo, Parallel Graph Algorithms, ACM Computing Survey 16 (1984) 319-348.

105. M.J.Quinn, Designing efficient Algorithms for Parallel computers, McGraw-Hill International Editions, Computer science series, 1987.

106. V.Ramachandran and J.H.Reif, An Optimal parallel algorithm for Graph planarity, in: Proc. 30th Annual IEEE Symposium on Foundation of Computer Science, (1989) 282-287.

107. S.Rengarajan and C.E. Veni Madhavan, Hamiltonian cycle in planar 3-trees, in: (S.C. De sarkal ed.) Proc. of the 3rd National Seminar on Theoretical Computer Science, India, 1993.

108. P.L.Renz, Intersection representations of graphs by Arcs, Pacific J. Math. 34 (1970), 501-510.

109. F.S.Roberts, Indifference graphs, in: Proof Techniques in Graph Theory, (F.Harary ed.), Academic press, New york (1971) 139-146.

110. F.S.Roberts, Discrete mathematical model with applications to social, biological and environmental Problems, Prentice Hall, Englewood cliffs, NJ.

111. D.Rose, Triangulated graphs and the elimination process, J. Math. Annal. Appl. 32 (1970) 597-609.

112. D.Rose, A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations, in: Graph Theory and Computing, (R. Read (ed.)) Academic Press, New york (1972) 183-217.

113. D.Rose, Simple characterizations of k-trees, Discrete Math. 7 (1974) 317-322.

114. D.Rose, R.Tarjan and G.Luekar, Algorithmic aspects of vertex elimination on graphs, SIAM. J. Comput. 5 (1976) 266-283.

115. A.N.Samy, G.Arumugam, M. Paul Devasahayam, and C. Xavier, A recognition algorithm for the intersection graphs of internally disjoint paths in Trees, in: P.S.Thiagarajan ed., National Seminar on Theoretical Computer Science,(Madras, India, 1991), 169-178.

116. A.A.Schaffer, A faster algorithm to recognize undirected path graphs, Discrete App. Math. To Appear.

117. C.E.Shannon, Zero-error capacity of a noisy channel, Comp. Information Theory, IRE Trans., 3 (1956) 3-15.

118. Y.Shibata, On the tree representation of Chordal graphs, J. Graph Theory 12 (1988) 421-428.

119. D.R.Shier, Some aspects of Perfect elimination orderings in chordal graphs, Discrete Appl. Maths. 7 (1984) 325-331.

120. W.K.Shih, T.C. Chen, and W.Hsu, An $O(n^2 \log^2 n)$ Algorithm for the Hamiltonian cycle problem on Circular-arc graphs, SIAM J. Comput. 21 (1992) 1026-1046.

121. Y.Shiloach and U.Vishkin, An O(log n) parallel connectivity algorithm, J. Algorithm 3 (1982) 57-67.

122. P.Sreenevasa kumar, Algorithmic and structural results on chordal graphs, PH.D. Thesis, IISc Bangalore, 1990.

123 P.Sreenevasa kumar and C.E.Veni Madhavan, A new class of separators and planarity of chordal graphs, Lecture Notes in Computer science 405 (1989) 30-43.

124. R.Sridhar and S.S.Iyenger, Fast Parallel Algorithms for recognizing strongly chordal, ptolemaic, and block graphs, Int. Con. Parallel Processing (1990), III-141-III-144.

125. S.K.Sukla, On Automorphism groups of Graphs, Ph.D. Thesis, IIT Kanpur, 1979.

126. L.Sun, Two classes of Perfect graphs, JCT B 53 (1991) 273-292.

127. M.M.Syslo, Triangulated edge intersection graphs of paths in a tree, Discrete Math. 55 (1985), 217-220.

128. R.E.Tarjan, Maximum cardinality Search and chordal graphs, Unpublished lecture Notes, Stanford University, 1976.

129. R.E.Tarjan and M.Yanakakis, Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs, SIAM. J. Comput. 13 (1984) 225-331.

130. R.E.Tarjan, Decomposition by clique separators, Discrete Math. 55 (1985), 221-231.

131. R.E.Tarjan and U.Vishkin, An efficient parallel biconnectivity algorithm, SIAM J Comp. 14 (1985) 862-874.

132. W.T.J. Trotter, A Note on triangulated graphs, Notices Amer. Math. Soc. 18 (1971) 1045 (A).

133. L.E.Trotter, Jr., On Line perfect graphs, Math. Program. 12 (1977) 255-259.

134. A.C.Tucker, The Strong Perfect Graph conjecture and an application to a municipal routing problems, in: Graph Theory and application, Lecture Notes on Math. Springer verlag, 303 (1972) 297-303.

135. A.Tucker, The Strong perfect graph conjecture for planar graphs, Canad. J. Math. 25 (1973) 103-114.

136. A.C.Tucker, Perfect graphs and an application to optimizing municipal services, SIAM Rev. 15 (1973) 585-590.

137. A.Tucker, Critical perfect graphs and Perfect 3-chromatic graphs, JCT B 23 (1977) 143-149.

138. A.C.Tucker, On Berge's Strong perfect graph conjecture, Ann. N.Y. Accd. Sci, 319 (1979) 530-535.

139. J.R.Walter, Representations of chordal graphs as subtrees of a tree, J. Graph Theory 2 (1978), 265-267.

140. H.Weiland, Finite permutation group, Academic Press Inc., New York, 1964.

Date Slip

This book is to be returned on the date last stamped.